# SPECTRUM

# VILÁG

- WHITE LIGHTNING
- GÉPI KÓD TANFOLYAM
- PIYAMARAMA/ BAT MAN
- HW: VIDEO KIMENET
- SW: ANIMÁCIÓ
- PAINTBOX




ELSŐ RÉSZ

# B E J E L E N T K E Z É S



Megkésve bár, de talán még időben kerülhet ez a kiadvány azoknak a kezébe, akik szabad idejüknek javarészét legszívesebben Spectrumuk társaságában töltik.
Szinte minden - a számítástechnikával foglalkozó - kiadvány hasábjain találkozhattunk olyan olvasói levelekkel, melyek rendszeres, sokoldalú és nagy menynyiségű információt igényeltek Sinclair-számítógépek felhasználásához, legyen az játék, hw/sw-trükkök és még sorolhatnánk...

Most jövünk mi !

Célunk kettős: egyrészt megfelelni az előzőekben leírt követelményeknek, másrészt viszonylag olcsó kiadvány életre hívása - nemcsak amíg a készlet tart (hiszen a téma kifogyhatatlan), hanem amíg igény van rá.
Terveinkről: az első két rész készen áll, a sorozat további darabjait szeretnénk az igényeknek megfelelően alakítani, sőt lehetőséget biztosítunk arra, hogy az olvasó íróvá váljék, bármilyen ötletét, írását szívesen fogadjuk.
Lehetőségeink: több részesre tervezett sorozatunk megjelenéséhez szükséges anyagi feltételeink 2 rész erejéig biztosítottak. A továbbiak a piaci sikerektől függnek, ezért már most elnézést kérünk az Olvasótól, ha netán vállalkozásunk gazdasági kudarcba fulladna, és emiatt megkezdett sorozataink befejezetlenül maradnának.

Adja a Spectrumosok lelkes és nagy táborának "istene", hogy következő számunk szerkesztői üzenetek c. rovata ne "KIJELENTKEZÉS" címmel jelenjen meg...

Felelős kiadó: Rucz Lajos

Címünk: "Spectrum Világ" Budapest 3.
Postán maradó
1300

Eng. szám: KF 56890
ISBN ÖK: 963 500 667 5
ISBN: 963 500 668 3

Készült: EGRESSY GÁBOR MUV. HÁZ, ÓCSA

# SIR CLIVE SINCLAIR TÜNDÖKLÉSE ÉS...

1940-et írtak, a németek teljes gőzzel törtek előre Európa nyugati része felé, a Brit-szigetek irányába. Ekkoriban egy Richmond nevű kisvárosban élt Bill Sinclair és felesége, Thora. Elsőszülött gyermeküknek, Clive-nek az élete kezdetben a háború szörnyűségei miatt elég keserves volt. Richmondi házukat lebombázták, új helyre kellett költözniük. Clive gyermekkori beceneve "Bambino" - amely a mai napig rajta maradt - onnan ered, hogy a születési anyakönyvi kivonatának betűjele: "BAMB". Clive gyermekkorában már kiderült, hogy kivételes képességekkel rendelkezik: három éves korában már felnőtthöz méltó hangsúllyal és kifejezőképességgel beszélt.

10 éves korában tanárai külön képzésre javasolták, ekkor már IQ mutatója az átlagosnál jóval magasabb volt. A fiatal Clive jó "koponyával" lett megáldva, 15 évesen képes volt a legbonyolultabb matematikai problémákat is a legrövidebb úton és idő alatt megoldani. 17. születésnapjára Clive a Weybridge-i St.George Iskolában folytatott tanulmányai mellett megszerkesztett egy mini-rádiót, amely nem volt nagyobb egy hagyományos öngyújtónál. Nem sokkal később egy újságíró megjósolta, hogy a Sinclair név még ismert lesz az emberek előtt. Ettől kezdetben féltek a szülők, de váratlanul Clive első unokatestvére beváltotta a jóslatot, területi, majd nemzeti vívó ill. atlétikai sikereket ért el - persze az újságíró nem így gondolta -. Clive nem volt sportember, bár indult a londoni és new york-i maratoni futóversenyen is, de számottevő sikereket a sport terén nem sikerült elérnie.

A "csend" csak addig tartott, amíg Clive meg nem alapította a személyi számítógépfejlesztéssel foglalkozó kísérleti kutatóközpontját, amely a "Sinclair Research Ltd." nevet kapta. Sok kísérlet, kutatómunka eredményeképpen született meg az első prototípus, melynek Clive a családi nevet adományozta. Ettől a pillanattól kezdve indult el a Sinclair név a világhír felé a számítástechnikai körökben. Clive "meredek" dologra szánta el magát: tetszetős dobozba kerülő, figyelemfelkeltő csomagolású gép mintadarabjait - mindennapos egyszerűséggel - szétosztogatta



utcai járókelők között. Ez is tükrözte kreatív gondolkodásmódját. Pályafutásának felfelé ívelő szakasza itt kezdődött. Rövidesen két megtisztelő cím is elterjedt róla: "Az év üzletembere" cím védője és "Az évtized számítástechnikai személyisége" lett.

Az első produkciója a ZX-80 volt, amelyet hamar követett utódja, a ZX-81. Az emberek szinte el sem hitték, hogy Z-80 mikroprocesszorral működő gépet 100 Font alatti áron is megvehetnek, hiszen ezt eddig 500 Font alatt nem tehették meg. A nagy siker láttán amerikai megrendelők is érkeztek. Igaz csak némi módosítással vették át a ZX-81 alapgépet, de TIMEX néven a gép hamarosan a tengerentúlon is ismert lett.

Elérkezett a "robbantás" éve: 1982-ben piacra került a SPECTRUM. Óriási Sinclair dömping indult meg: a gép először 16, majd ezzel párhuzamosan 48 K-s változatokban a világ minden részén gombamódra kezdett szaporodni.

Anyja szemében csak akkor vált igazán megalapozottá az újságíró egykori kijelentése, amikor Clive a munkásságáért elnyerte a lovagi címet. A lovagi cím átadására 1983-ban - a királynő születésnapjának tiszteletére megrendezett ünnepségen - került sor. Akkor a "Sinclair Research Ltd." főnöke már milliomos volt, tevékenységével meghatározta az egész szigetország számítógépes gondolatvilágát.

# ...BUKÁSA?!

Egy újabb év elteltével megszületett a QL, amely már nem volt akkora siker, mint elődje, de - immáron Sir - Clive még mindig nem panaszkodhatott. Párhuzamosan viszont új dologba vágott bele: a C5 típusú elektromos autó kifejlesztésébe. Ekkor még nem sejthette, hogy ez meghatározza későbbi pályafutását. Megrendelései voltak, s a kísérleti időszak után a C5 már futott az utakon.

A QL átmeneti kudarca és a felhasználók igénye alapján a tervező stáb - Sir Clive vezetésével - úgy gondolta, hogy a lehetséges legjobb megoldás, ha egy ötvözött gépet hoznak létre. Így született meg a Spectrum 128K. Ugyanebben az évben 1985-ben legyártották az 1 milliomodik Spectrumot is (amelynek közben módosított "+" jelű változata is forgalomba került). A jubileumi példány ünnepi külsőt kapott: hófehér dobozban, aranyozott billentyűkkel adta át a gyártó - ezrek jelenlétében - Sir Clive-nak a Skót Nemzeti Csarnokban.

Elérkezett 1986, ami rosszul indult: a rendelések visszaestek, a központi vevők raktárai "dugig" voltak a Sinclair gépekkel, nem beszélve az egyéni érdeklődés csökkenéséről. Ez kihatott a termékárak csökkenésére is. A C5 típusú elektromos autó konstrukciós, valamint üzlettaktikai hibáiból adódóan a cég a versenytárgyalásokon alulmaradt, sőt az eredeti rendelők is visszamondták igényléseiket. A történtek alapján a céget nagy anyagi veszteség érte, s ebből a csávából még a 128-as megjelenése sem jelentett kiutat. Sir Clive eladósodott, az egyedüli megoldás a cég áruba bocsátása lett, és - az ekkor már felfutóban lévő - AMSTRAD cég "főnöke" Alan Sugar szinte nevetséges kikiáltási árért megvásárolta a Sinclair Research Ltd.-t.

Tévedés lenne azt gondolnunk, hogy evvel a Sinclair gépek történetének vége. Ennek több oka is van: a meglévő géppark ettől még éppen olyan sikeres mint régen, a software-fejlesztők várhatóan a jövőben is készítenek software-eket Sinclair gépekre.

Az AMSTRAD cég a beolvadást követő időszaktól újabb "közös" típusok megjelenését segíti elő. Az AMSTRAD-SINCLAIR Spectrum 128 +2 gép 128 K-s, keménybillentyűs, beépített AMSTRAD magnóval rendelkezik, a legújabb +3 típus pedig már mágneslemezegységet is tartalmaz. A legújabb hírek szerint Sir Sinclair is új gépet fejlesztett ki: a Z-88-at (az említett típusokról a következő számokban még lesz szó), amelyet a Cambridge Electronic Ltd. forgalmaz. A gép nem viselheti a Sinclair nevet, hiszen azt az AMSTRAD megvette.

Összességében megállapítható, hogy a Sinclair cég gondjai nem okoztak zavarokat a felhasználók körében, sőt lehet, hogy mindez a fejlesztés előnyére vált: napjainkban a software-cégeknek még mindig az egyik legjobb befektetése Sinclair gépekre programot írni!

## TOP 10

1987 - AUGUSZTUS

| | | |
|---|---|---|
| 1. | FEUD | - Bulldog |
| 2. | GAUNTLET | - US Gold |
| 3. | ROBIN OF THE WOOD | - Odin |
| 4. | PAPERBOY | - Elite |
| 5. | OLLI AND LISA | - Firebird |
| 6. | SHORT CIRCUIT | - Ocean |
| 7. | ARNHEM | - CCS |
| 8. | PIYAMARAMA | - Mikro-Gen |
| 9. | TIR NA NOG | - Gargoyle Games |
| 10. | BAT MAN | - Ocean |

# JÁTÉKÚJDONSÁGOK A PIACON

## NINJA - Mastertronic

Megélénkült a karateszimulációk gyártása az angol játékpiacon: az AVENGER (WAY OF THE TIGER II.), az AVENGING ANGEL (SABOTEUR II.) és az EXPLODING FIST II. sikere után a Mastertronic is egy új karateszimulációval rukkolt ki. Akik szeretik az ilyen típusú játékokat, meg lehetnek elégedve vele: a grafika jó, a játék izgalmas. Az akció több szinten zajlik, a különböző szinteken egyre több ellenséggel találkozhat a játékos. A szintek között a képernyő tetején látható átjárókon lehet feljutni.

## MARTIANOIDS - US Gold

Mióta az ULTIMATE cég beolvadt a US GOLDba se vége, se hossza azoknak az új játékoknak, amelyek a klasszikus KNIGHT LORE-stílusban zajlanak (pl. ilyen a később bemutatásra kerülő BATMAN). A MARTIANOIDS egy szokásos mászkálós játék, amelyben egy robottal kóválygunk egy labirintusban ahol különféle tárgyakat kell gyűjtenünk. Ezt barátságtalan élőlények akadályozzák, amelyek fogyasztják a robot telepeit (energiáját). Az ellenségeket és a mozgásunkat gátló falakat lézerrel semmisíthetjük meg. A grafika jó, akik szeretik a mászkálós játékokat, biztosan megkedvelik ezt is.

## SYGMA 7 - Durell

Az URIDIUM nemes hagyományait követő űrhajós/lövöldözős játék, kitűnő zenével és grafikával. Az első pályán az űrben kell harcolnunk az ellenséggel, a második pályán elhagyjuk az űrhajónkat és pontokat gyűjtünk. A harmadik pályán egy gömbbel vagyunk. Ha nem KEMPSTON joysticket választunk, definiálhatjuk a vezérlő billentyűket.

## STRIPPER - Individual

A SAMANTHA FOX nyomdokain készült játék. Normál ötlapos póker, hagyományos játékszabályokkal: két licitálási lehetőség van (közöttük cserével), a limit 20 font. Minden pályán más hölggyel mérhetjük össze tudásunkat (és szerencsénket). A lányok tudása egyre jobban nő, a 4. pályán lévő hölgy például olyan szemérmetlenül blöfföl, mint egy igazi hamiskártyás. A játék 90-90 fonttal indul, ha elvesztünk mindenünket, elvész egy életünk (induláskor 5 van). Akik nem tudnak pókerezni, a játék főmenüjéből kérhetnek - angol nyelvű - instrukciókat. Az irányítás definiálható, alaphelyzetben a CURSOR-joystick billentyűinek felel meg.

## UCHI MATA - Martech

Jó grafikával és animációval kivitelezett judoszimuláció. A harc - természetesen - az ellenfél legyőzéséért folyik, játszható a gép ellen vagy két játékosal. A gyakorló üzemmódban megismerkedhetünk a játék közben alkalmazható mozgás- és dobásformulákkal. Ezeket a program a képernyő felső részén jelzi az adott formula neve melletti szimbólum színezésével.

## SCALEXTRIC - Leisure

Rendkívül jó grafikával megrajzolt autóverseny-szimuláció. Két játékos játszhatja, de lehet partnerünk a gép is, ha a "2 PLAYER" kérdésre ENTER megnyomásával válaszolunk. Ha a gép ellen kívánunk játszani, be kell állítanunk a nehézségi fokozatot is. Definiálhatóak a gyorsítás-lassítás/gáz/fék funkciókat kiváltó billentyűk. A programban megtalálható jónéhány Forma-I versenyekből ismert pálya, de tervezhetünk saját pályát is, amit az újrafelhasználás céljából kimenthetünk. A programban a játékosok egy kettéosztott képernyőn figyelhetik a saját tevékenységüket és az ellenfél mozdulatait.

# PIYAMARAMA




A Mikro-Gen cég Wally-játékai egyedülálló stílust képviselnek az arcade-típusú játékok között: az általuk forgalmazott programok nagy részén mintegy védjegyként vonul végig, hogy a játéktérben szétszórt tárgyakat egy bizonyos logika alapján a megfelelő helyre kell vinni, tehát a játékok sikeres megoldásához nem csak ügyes kezekre, hanem egy jó adag asszociációs készségre is szükség van. Az EV'RYONE'S A WALLY című játékával a Mikro Gen olyan hatalmas sikert ért el, hogy újabb és újabb Wally kalandokat feldolgozó játékokat dobott piacra. Ezek közül az LSI Atsz. 1987-ben megjelent "Spectrum III." című könyvében már megismerkedhettünk a negyedikkel, a 3 WEEKS IN PARADISE-al. A sorozat második tagja a most ismertetésre kerülő PIYAMARAMA (második rész), talán a legnagyobb sikert elért Wally játék.

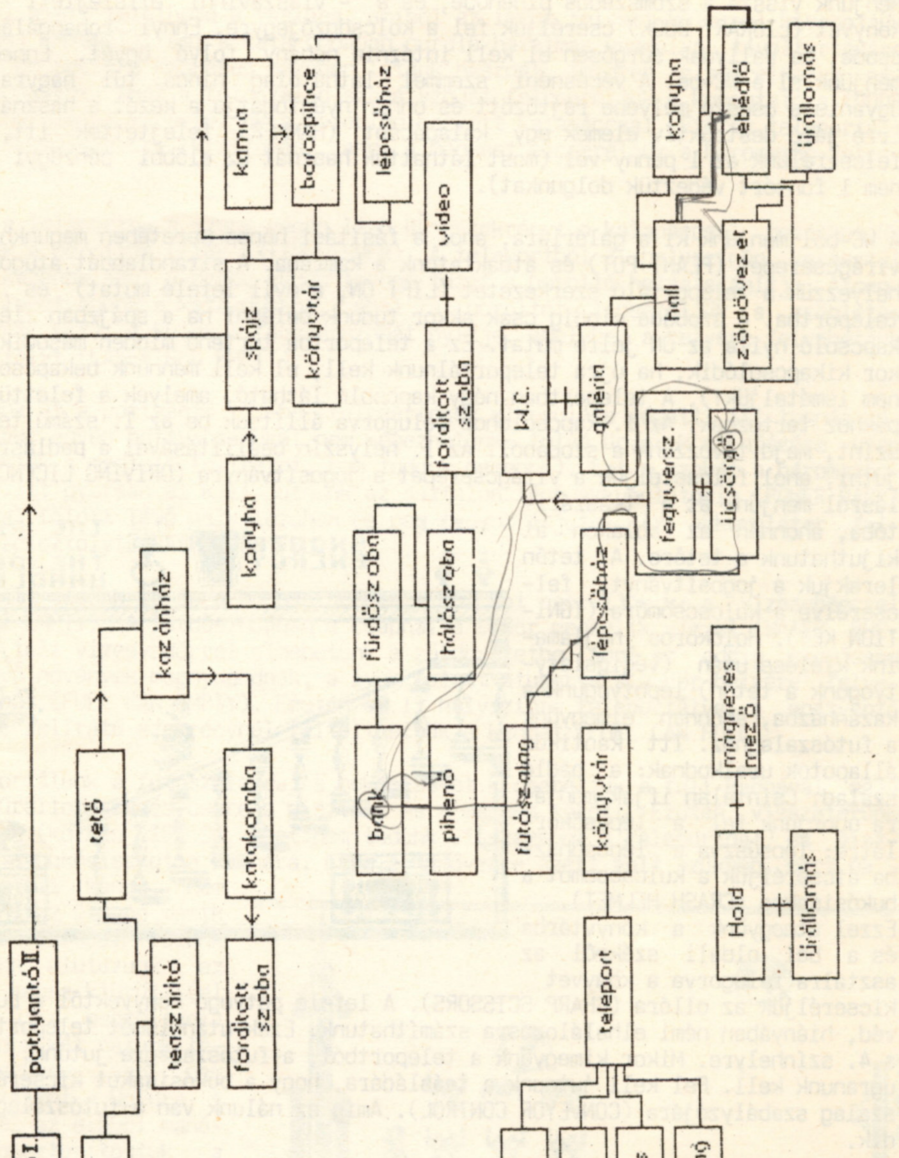
Betöltés után a program elmuzsikálja a "Popcorn" (Pattogatott kukorica) című világsláger néhány taktusát, majd egy gomb megnyomása után a főmenüben beállíthatjuk a játék irányítását (SINCLAIR/KEMPSTON-joystick vagy billentyűzet definiálva) és ENTER billentyűvel indíthatjuk a játékot. A játékot a BREAK megnyomásával bármikor megszakíthatjuk és visszatérhetünk a főmenübe. Mielőtt elindulnánk, tanulmányozzuk a térképet (a leírásban a szobák nevére fogunk hivatkozni). A térképen nyíllal jelöltük az egyirányú utcákat és áthúzással azokat a helyeket, ahova csak egy bizonyos tárgy birtokában tudunk bejutni.

A játék egy házban játszódik, amelynek tulajdonosa, Wally édesded álomba merülve hortyog. Tudatalatti lénye azonban kilép testéből és elindul, hogy elhozza szendergő tulajdonosának az ébresztőóra kulcsát, miáltal reggel felébred és nem késik el a munkából (mint az jó szokása). Wally házában kiváló káosz uralkodik, ezért ez a feladat - a legenyhébb kifejezéssel élve - kissé körülményes mulatság. A házban a mászkáló grillcsirkéktől kezdve a repkedő indián csatabárdokig rengeteg ellenséggel találkozunk, amelyek érintésére a képernyő tetején, a pohárban lévő életerőnk (SNOOZE ENERGY) egy egységnyit fogy. Ha a pohár kiürült Wally - érdemei elismerése mellett - a mennybe megy és a bal felső sarokban látható életeink közül egy elvész. A házban egyébként hihetetlen történések adódnak: a padlóból hatalmas kezek nyúlkálnak, az ajtókon való átkelési kísérletekkor egy bokszkesztyű teljesen spontán módon orrba ver minket (sebaj, elmondhatjuk, hogy kesztyűs kézzel bánnak velünk).

Felhívjuk a figyelmet arra, hogy a leírásban minden tárgy felvételére - a lelőhelylyel együtt - hivatkozni fogunk, ha útközben valamilyen más tárgy akad az utunkba azt át kell ugorni, nem szabad felvenni! Kezdődjék hát a játék...

A játék kezdetén a hallban vagyunk. Menjünk át az ebédlőbe és vegyük fel az egyfontos érmét (POUND COIN - ez egy fontos egyfontos), majd az első emeleti ücsörgőben a kölcsönző cédulát (LIBRARY TICKET). Ezután menjünk fel a második emeleten lévő bankba (a futószalagon egyenlőre csak előre ugrálva tudunk átjutni). A bankban - rövid lejáratra - elhelyezzük az egyfontosunkat és rögtön le is vesszük a kamatot egy 1 pennys képében (PENNY).

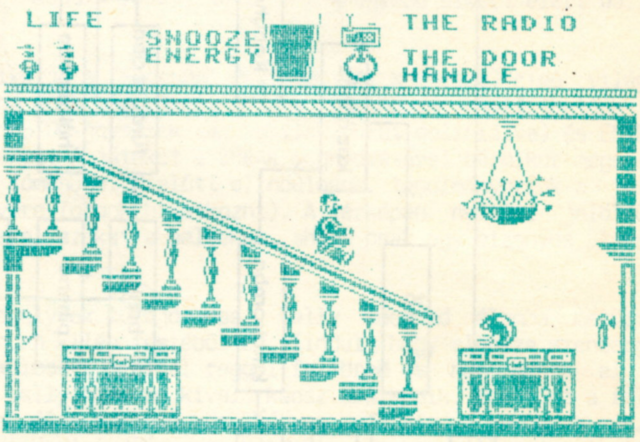
# PIYAMARAMA


pottyantó I.
padlás
pottyantó II.
teasz árító
tető
kaz ánház
fordított szoba
katakomba
konyha
spájz
kamra
könyvtár
borospince
lépcsőház
video
fordított szoba
fürdősz oba
hálósz oba
bank
pihenő
W.C.
galéria
futósz alag
könyvtár
lépcsőház
teleport
padlás
könyvtár
űrállomás
futósz alag
fegyversz
üsszögő
hall
zöldövezet
konyha
ebédlő
űrállomás
mágneses mező
Hold
űrállomás

# PIYAMARAMA

Menjünk vissza a szomszédos pihenőbe, és a - visszavinni elfelejtett - könyvtári könyvet (LIBRARY BOOK) cseréljük fel a kölcsönzőjegyre. Ennyi rohangálás után nem csoda, ha Wallynak sürgősen el kell intéznie néhány folyó ügyét. Ennek érdekében menjünk el a WC-be. A vécésnéni szemmel láthatólag nincs túl nagyra rangjával, ugyanis a csésze mélyébe rejtőzött és onnan nyújtogatja a kezét a használati díjért. Erre járó destruktív elemek egy kalapácsot (HAMMER) felejtettek itt, amit most felcserélünk az 1 penny-vel (most láthattuk hasznát az előbbi pénzügyi manővernek: nem 1 fontért végeztük dolgunkat).

A WC-ből menjünk ki a galériára, ahol a fásítási hónap keretében magunkhoz vesszük a virágcserepet (PLANT POT) és átbaktatunk a kamrába. A strandlabdát átugorva, üzembe helyezzük a teleportáló szerkezetet (LIFT ON, a nyíl lefelé mutat) és elmegyünk a teleportba. A szobába mindig csak akkor tudunk belépni ha a spájzban lévő LIFT ON kapcsoló nyila az ON jelre mutat. Ez a teleportba történő minden második belépésünkkor kikapcsolódik, ha újra teleportálnunk kell, el kell mennünk bekapcsolni (többet nem ismételjük!). A teleportban négy kapcsoló látható, amelyek a felettük lógó lámpákhoz tartoznak. Az 1. kapcsolóhoz felugorva állítsuk be az 1. számú teleport-helyszínt, majd távozzunk a szobából. Az 1. helyszín beállításával a padlásra tudunk kijutni, ahol felcseréljük a virágcserepet a jogosítványra (DRIVING LICENCE). A padlásról menjünk át a teaszárítóba, ahonnan az ablakon át kijuthatunk a tetőre. A tetőn lerakjuk a jogosítványt, felcserélve a kulcscsomóra (IGNITION KEY). Holdkóros hajlamaink kiélése után (végigslattyogunk a tetőn) lepotyogunk a kazánházba, ahonnan elmegyünk a futószalaghoz. Itt kaotikus állapotok uralkodnak: a padló szalad. Csintalan ifjak módjára ugorjunk fel a lépcsőkorlátra: lecsúszva a lépcsőházba átcseréljük a kulcscsomót a bukósisakra (CRASH HELMET). Ezzel átmegyünk a könyvtárba és a bal oldali székről az asztalra felugorva a könyvet kicseréljük az ollóra (SHARP SCISSORS). A lefelé potyogó könyvektől a bukósisak megvéd, hiányában némi elhalálozásra számíthatunk. Ezek után ismét teleportálnunk kell, a 4. színhelyre. Mikor kimegyünk a teleportból, a futószalagra jutunk. Itt azonnal ugranunk kell. Fel kell jutnunk a teaságládára, hogy a bukósisakot kicseréljük a futószalag szabályzójára (CONVEYOR CONTROL). Amíg ez nálunk van a futószalag nem működik.




Most el kell jutnunk a kamrába, ahol a magasban egy kulcs található. Ezt ugrással nem tudjuk elérni. A bankban lévő HELP ON kapcsolót aktivizálva, a kulcs alatt egy teásláda jelenik meg, amiről a kulcs elérhető. A kapcsoló üzembe helyezése után, menjünk a kamrába, ahol a középen lévő hordót át kell ugranunk, hogy a szoba másik végében lévő léggömbhöz eljussunk. Ha ez sikerült (leeséskor újra HELP ON és csak azután vissza!), a lufit megérintve felrepülünk a felső részre. A lyukakat átugrálva eljutunk a teásládáig és kicseréljük az ollót a vasláda kulcsára (BOX KEY). Ezután potyogjunk vissza a földre és menjünk a konyhába.

# PIYAMARAMA

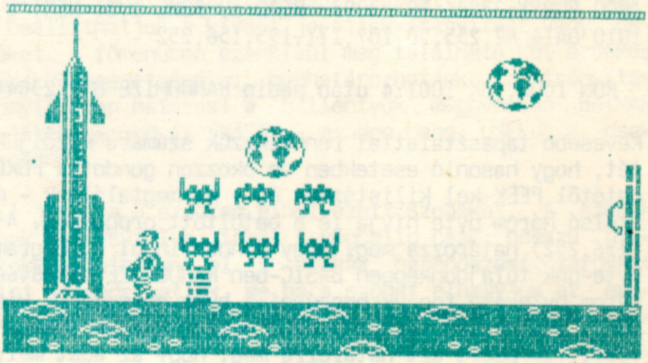
Itt az asztal alatt található a vasláda, amiből az imént felvett kulccsal kivehetjük a patkómágnest (MAGNET), ami a katakombában guruló hatalmas vasgolyó távoltartására szolgál, később pedig a mágneses mezőben látjuk hasznát. Imígyen felszerelkezve sétáljunk át a hallba, ahol a futószalag szabályzóját felcseréljük a kopogtatóra (DOOR HANDLE). A kopogtatót kicseréljük a galérián hagyott kalapáccsal. Ezután teleportálnunk kell a 3. helyszínre (űrállomás), ahol a mágnest elcseréljük a háromszög-kulcsra (TRIANGLE KEY).

Visszatérve a teleportba a szoba végén lévő üvegszekrényt a kalapáccsal betörhetjük és kivehetjük a poroltót (FIRE EXTINGUISHER). A teleportból 2. helyszínen keresztül elmegyünk a kazánba ahol a szoba végén lobogó tüzet a poroltó hatástalanítja. Átmenve a tűzön, lepotyogunk a katakombába, ahol a poroltót lecseréljük a szögletes kulcsra (SQUARE KEY). A kulcs a fegyverszobát nyitja, amelyben a helyére felvehetjük a lézer-pisztolyt (LASER GUN EMPTY).

Ezután menjünk át a pottyantó II.-be (félreértések elkerülése végett: itt tonnás súlyok potyognak, egyéb pottyantásokat nem végezhetünk: folyó ügyeink elintézése után a WC-re kirakták a megtelt feliratot, nem tudunk bemenni). Ezt a szobát a háromszögkulcs nyitja.Itt gyorsan el kell jutnunk az asztalon lévő akkumulátorig (POWER PACK) mert az asztal fölött lévő súly lezuhan és nem tudunk átjutni. Az akkumulátor feltölti a lézerpisztolyt (LASER GUN FULL).

Az asztal alatt essünk le és a borospincében a kulcsot cseréljük ki a vizesvödörrel (WATER BUCKET EMPTY). A fürdőszobában a csapnál a vödör megtelik vízzel (WATER BUCKET FULL). A teli vizesvödörrel elmehetünk a zöldövezetbe, ahol az eddig szomjasan acsargó húsevő növények megnyugodnak, a lyuk felett átugorva a vödör helyére felvehetjük a kannát (FUEL CAN EMPTY). Ezután az 1. helyszínre teleportálva, a pottyantó I.-ben a FUEL feliratú szekrénynél feltölthetjük a kannát (FUEL CAN FULL).

Ezután teleportálunk a futószalagra, a bukósisakot kicseréljük a teli kannával, majd teleport az űrállomásra és csere a pisztoly és a mágnes között. Ezután visszatérünk a futószalagra (közben LIFT ON, mert nem tudunk visszamenni a teleportba!) és viszszacseréljük a bukósisakot a kannára. Erre a manőverre azért volt szükség, mert ha nálunk van a teli kanna, nem tudunk átmenni az űrhajó másik oldalára, mert elrepül velünk a holdra. Ha elutazunk az űrhajóval, a kanna kiürül (el kell vinnünk újra megtölteni) és mivel mind a két nálunk lévő tárgyra szükség van, nem tudjuk feljuttatni a Holdra a mágnest. Mivel az előbbi manőverrel átcsoportosítottuk a tárgyainkat, a kanna újabb felvétele után a teleportból átmehetünk az űrállomásra és elrepülhetünk a bolygóra, ahol kicseréljük a mágnest a Holdkristályra (MOON CRYSTAL). Szálljunk vissza a rakétába és leszállás után cseréljük ki a holdkristályt a pisztollyal és menjünk el újra feltölteni az utazás közben kiürült kannát.



# PIYAMARAMA

Ha megtöltöttük a kannát újabb űrutazásra indulunk (úgy repkedünk már a Föld és a Hold között, mint valami füstifecske). A Holdon kicseréljük a kannát a mágnesre. Mivel nálunk van a feltöltött lézerpisztoly, a holdlakók felmenekülnek a magasba, felhasználva a mágnest át tudunk menni a szomszéd szobában lévő mágneses mezőbe. Itt a gombot megnyomva megszűnik a mező és felugrálva felvehetjük az óra felhúzóját (CLOCK KEY). Ezután visszatérünk a rakétába, leszállás után pedig elmegyünk a hálószobába, ahol belerakjuk az órába a kulcsot. A bűnös szerkezet rút csörömpöléssel megszólal..

Némi tűzijáték következik, majd a gép közli kézmeleg gratulációját: WALLY FELÉBREDT ÉS ÉLETÉBEN ELŐSZÖR PONTOSAN FOG A MUNKÁBA MEGÉRKEZNI. Egy billentyű megnyomása után megnézhetjük, hány lépéssel és hány százalékra tudtuk a játékot teljesíteni.

A PIYAMARAMA kicsit bonyolultabb logikával rendelkezik, mint a 3 WEEKS IN PARADISE, ahol egy lépésből logikusan következik az új lépés. A logika bonyolult volta miatt, nem kommentáltuk az olyan lépéseket, amelyek későbbi manővereket készítettek elő vagy könnyítettek meg. Ha valaki jobban elmélyed a játékban, úgyis magától rájön az összefüggésekre.

A játék többféleképpen is megoldható, de ezen a vonalon ajánljuk, hogy a lépések sorrendjét ne kavarják össze, főleg ne hagyjanak ki lépéseket. Játék közben észrevehetik, hogy van 6-7 olyan tárgy, amit semmire sem használtunk, fel sem vettünk. Hogyan kell ezeket a megfelelő helyen elhelyezni és ezzel száz százalékra teljesíteni a játékot? Ez már a Kedves Olvasó dolga.

A játékot kicsit nehézkes lenne 3 élettel teljesíteni, kézenfekvőbb örökélettel: a 43883. címre kell a kívánt életek számát (1-255) bepoke-olni. Illetéktelen (és tapasztalatlan) kezektől a program bizonyos fokig védett, tehát ismertetjük a POKE bevitelének a módját:

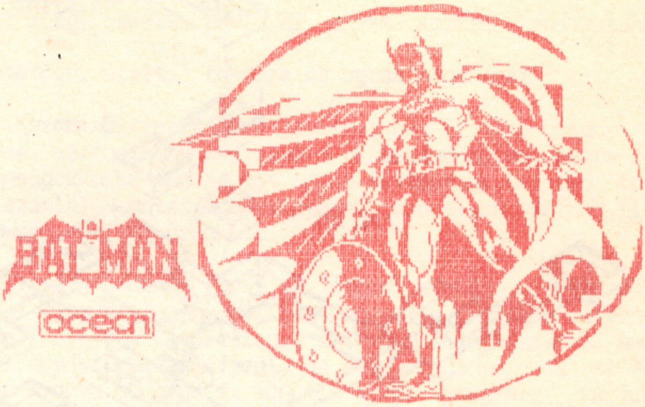
A BASIC-rész betöltése után BREAK-kel állítsuk meg a programot. LIST után a 0. sorban értelmetlen adatok tűnnek fel, amiből levonhatjuk a következtetést: a fejléc nélküli file-ok gépi kódú betöltője BASIC területen van. Ezt némileg módosítjuk:

```
1000 FOR i=23886 TO 23893: READ a: POKE i,a: NEXT i
1010 DATA 62,255,50,107,171,195,156,252

  RUN 1000, OK 1001:4 után pedig RANDOMIZE USR 23844
```

Kevesebb tapasztalattal rendelkezők számára közöljük a megvalósítás elméleti menetét, hogy hasonló esetekben ne okozzon gondot a POKE bevitele: a BASIC-terület kezdetétől PEEK-kel kilistázzuk és - ha megtaláltuk - megvizsgáljuk a töltő adatait: az utolsó három byte hívja le a betöltött programot. A 195 (JUMP) után álló két szám (156,252) határozza meg, hogy honnan indul a program (156 1+252 256), tehát utolsó 3 byte-unk tulajdonképpen BASIC-ben RANDOMIZE USR 64668 utasításnak felelne meg. Ezen három byte elé fogjuk beszúrni 5 byte hosszúságú kiegészítésünket: a második szám (255) jelenti a bevinni kívánt adatot (jelen esetben az életek számát), a 4-5. szám (107,171) pedig azt határozza meg, hogy az adat melyik címen fog elhelyezkedni. Ezt a következőképpen számolhatjuk: a második byte értékét megszorozva 256-tal és az eredményhez hozzáadva az első byte értékét kapjuk meg a címet, esetünkben a 43883-at a 107 és a 171 határozza meg. Utolsó lépésben lehívjuk a betöltő rutint.

# BAT MAN




A mikroszámítógépekre készült játékprogramok íróinak kimeríthetetlen ötletforrásai a képregények, angol nevükön comics-újságok. A képregényeket eredetileg a csökkent képességű gyermekek oktatásához használták, de a közöttük aratott hatalmas siker indokolttá tette azt, hogy a nagy közönség előtt is kipróbálják. Az ötlet kiválóan bevált, a társadalom ifjabb és idősebb rétege is osztatlan lelkesedéssel fogadta az újítást. A képregények diadalmenete napjainkban is tart: a piacon soha nem látott tömegű comics kapható (pl. Bécsben egy külvárosi boltban több, mint 2000 /!/ folytatásos képregény különféle számaiból válogathat a vásárló). A képregények a történet szempontjából a lehető legszélesebb skálát ölelik fel: a gyermekmeséktől a krimikig és a szépirodalom remekeiig szinte minden megtalálható. Külön kategóriát képeznek a - leginkább - 6-14 éves korosztálynak készült, "szuperemberek"-ről szóló sorozatok. Ezek között a legismertebb a SUPER MAN és a BATMAN.

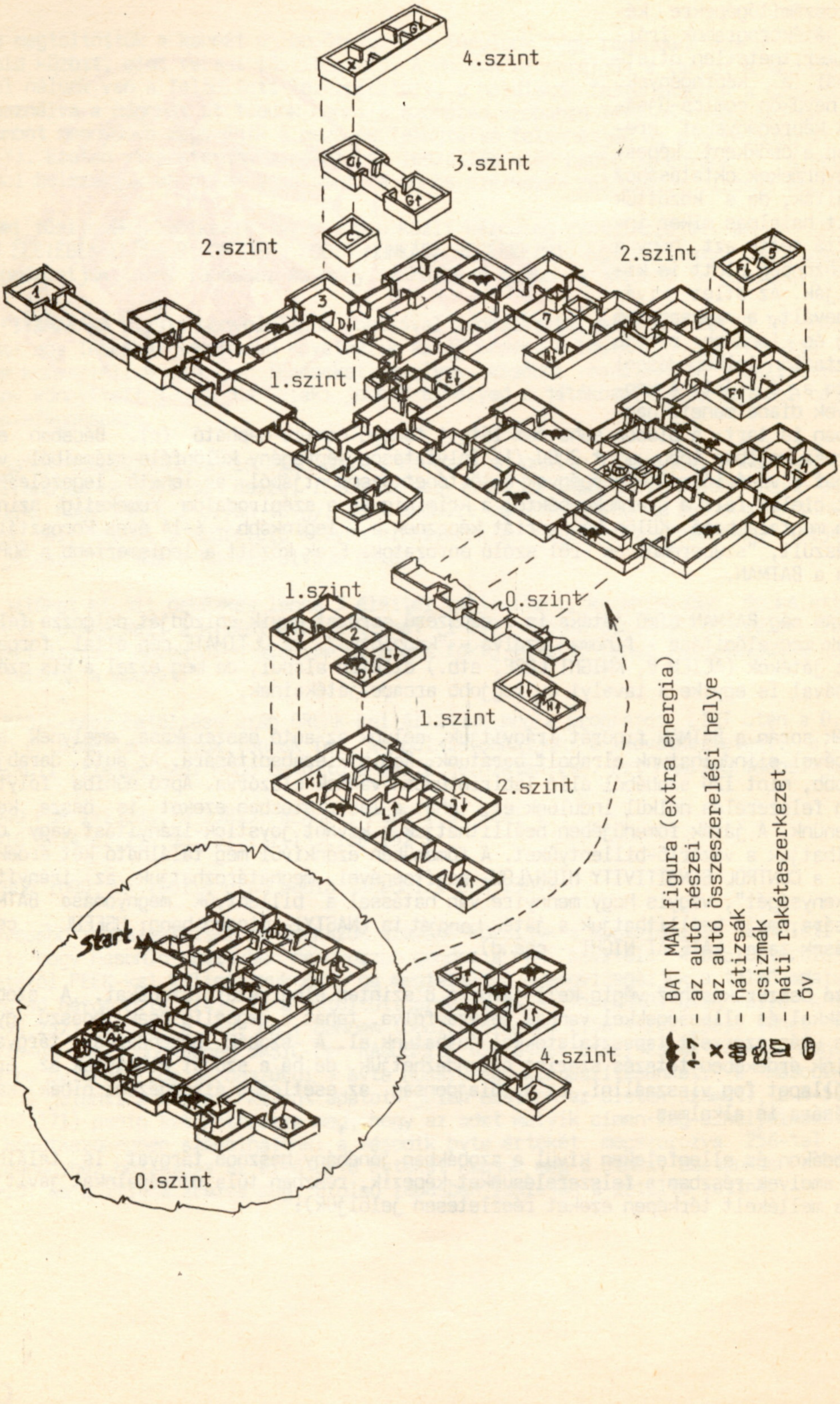
Az OCEAN cég BATMAN című játéka is a népszerű sorozat egyik epizódját dolgozza fel. A játék megvalósítása - finoman szólva - "koppintás", az ULTIMATE cég által forgalmazott játékok (ALIEN 8, KNIGHT LORE, stb.) ötletén alapul, de még ezzel a kis szépséghibával is egyike a tavalyi év legjobb arcade-játékainak.

A játék során a BATMAN figurát irányítjuk, célunk az autó összerakása, amelynek segítségével elindulhatunk elrabolt barátunk, Robin kiszabadítására. Az autó darabjai egy több, mint 120 szobából álló labirintusban vannak elszórva. Apró műhiba folytán minden felszerelés nélkül indulunk el, tehát a labirintusban ezeket is össze kell gyűjtenünk. A játék főmenüjében beállíthatjuk a kívánt joystick-irányítást vagy definiálhatjuk a vezérlő-billentyűket. A főmenüben ezenkívül még található két érdekes opció: a CONTROL SENSITIVITY HIGH/LOW segítségével meghatározhatjuk az irányítás "érzékenységét", vagyis hogy mennyire van hatással a billentyűk megnyomása BATMAN mozgására, majd beállíthatjuk a játék hangját is (NASTY - minden hang; USEFUL - csak a lépések zaja; LATE AT NIGHT - csend).

Az autó összerakásához végig kell járni a 8 szinten elhelyezett szobákat. A szobák csapdákkal és ellenségekkel vannak telezsúfolva, tehát a végcélig csak hosszú gyakorlás után szerzett tapasztalatokkal juthatunk el. A szobák berendezési tárgyait céljaink érdekében tetszés szerint átrendezhetjük, de ha a szobát elhagyjuk az eredeti állapot fog visszaállni. Ez a tulajdonság, az esetleges átrendezési hibák korrigálására is alkalmas.

A csapdákon és ellenfeleken kívül a szobákban jónéhány hasznos tárgyat is találhatunk, amelyek részben a felszerelésünket képezik, részben tulajdonságainkat javítják fel (a mellékelt térképen ezeket részletesen jelöljük):

# BAT MAN


4.szint
3.szint
2.szint
2.szint
1.szint
1.szint
0.szint
1.szint
2.szint
Start
4.szint
0.szint
BAT MAN figura (extra energia)
1-7 – az autó részei
X – az autó összeszerelési helye
– hátizsák
– csizmák
– háti rakétaszerkezet
– öv

# BAT MAN

- hátizsák: a tárgyak összegyűjtéséhez szükséges
- csizmák: csak felvételük után tudunk ugrálni
- "háti távolságleküzdő szerkezet": egy rakéta, amellyel ugrás közben repülhetünk illetve fordulhatunk
- öv: nagyobb lendületet tesz lehetővé, ezáltal fokozza az ugrás nagyságát

Tulajdonságainkat a kis BATMAN figurák megszerzésével javíthatjuk. Ez több dolgot eredményezhet: sebesség megnövelése, sérthetetlenség, dupla magasságú ugrás, extra élet. Természetesen ezek a tuladonságok csak megszabott ideig érvényesek (az erő fogyását a bal alsó sarokban lévő számlálók mutatják). Vigyázat, van olyan BATMAN-figura is, amely elveszi az addig megszerzett erőnket!

Az akció a 0. szinten, a START feliratú pontról indul. A játékot mindig a hátizsák felvételével kezdjük, hogy a többi tárgyat legyen hol raktároznunk. Ezután a csizmákat kell megszereznünk. A szobába lépve egy futószalagra pottyanunk, vitessük el magunkat a szoba alsó sarkába, ahol egy ellenség kóricál. Ha elég közel megy el előttünk, a 'fel'-gombot folyamatosan nyomva menjünk el mögötte a 3. beugróig. Várjuk meg amíg az ellenség elmegy, engedjük el a gombot és szaladjunk el a csizmákért.

Ezután térjünk vissza előbbi helyünkre, várjuk meg, míg az ellenség elmegy és mielőtt megfordulna távozzunk a szobából arra, amerről jöttünk. A csizmákat megszereztük, most már tudunk ugrálni, felugorhatunk a futószalagra. Ezen is csak ugrálva juthatunk előre, lépegetve egy helyben maradunk.

A csizma megszerzése után induljunk el a rakétákat is begyűjteni. A szobában a buborékot a zsámoly, majd a rakéták felé kell lökni és azután a bal oldalon lévő sámlikon a buborékkal egy magasságba kell ugrálnunk. Mivel ugrásaink most még elég kicsik ezért teljesen ki kell menni a sámli szélére (helyzetünk megítélésében segít a köpenyünk: addig mehetünk lepottyanás nélkül, ameddig még érinti a padlót). Ugráljunk be a rakéták melletti téglarakásra, menjünk ki a széléig és nyomjuk meg a 'tárgy felvétele'-gombot kétszer. A kijutás hasonlóképpen történik a bejutáshoz.

Mielőtt befejeznénk ténykedésünket ezen a szinten, gyűjtsük be az extra erőket.
A következő szintre a "B" betűvel jelzett szobából juthatunk át. A szobában két fazék és egy kancsó látható, amelyeket a 'tárgy felvétele'-gombbal és némi mozgolódással lépcsőzeteen összeépíthetünk. Remekünket toljuk be a jobb oldali sarokba úgy, hogy a futószalag mellett álljon. Ugorjunk fel az alsó fazékra, majd az 'ugrás' és a 'tárgy felvétele' gombok együttes használatával ugorjunk a következő fokra. Ezután tegyük le a fazekat, ugorjunk fel a futószalagra, majd a liftező kockára rápottyanva átmegyünk a következő szintre.

Az egyes szobák teljesítésének technikáját nem kívánjuk részletezni (ezt a játékosra bízzuk), mert a játék folyamán a felmerülő nehézségeket a 0.szint szobáinak imént ismertetett "logikája" alapján kell megoldani.

Ha a szobákban sikerül összegyűjtenünk az autó összes részét (7 db), a 4. szinten lévő, a térképen "X" betűvel jelölt szobába kell jutnunk, ahol BATMAN beszáll az autójába és elindul Robin megmentésére. Ennél a lehetőségnél valószínűbb, hogy idő előtt elhalálozik az összes emberkénk. Ha a gép jó hozzánk, még egy lehetőséget (DOGS LIFE) kapunk az akció teljesítésére.

A játékban találkozhatunk felvehető kristálygömbökkel. Ha egy ilyet felveszünk, a gép megjegyzi a szobát és a játék befejeződése után, a főmenü OLD GAME opciójának választásával az új játékot ebből a szobából kezdhetjük.

Örökélet: A 36800. címen zérust kell elhelyeznünk !

# MÁSOLÓPROGRAMOK

## COPIER FM-3

Régi "harcos" már a szakmában, de máig is az egyik legjobb copy-programnak számít. Memóriakapacitása 41535 byte, a betöltött file-okat darabonként vagy auto-save módban is kimenthetjük. A kimentés elindításakor "csiripelő" hangjelzést tehetünk a file-ok elejére, ami gyorskeresővel rendelkező magnók esetében roppant praktikus dolog: a BASIC-betöltőt hangjelzéssel megjelölve nagyon könnyen megtalálható a program eleje.

Betöltés után azonnal LOAD üzemmódba áll. A képernyőn több feliratot láthatunk:
-a RADIX jelzi a betöltött byte-ok hosszának a kijelzési módját, ami hexadecimális vagy decimális formában történhet. A kijelzést a SYMBOL SHIFT és a 4 billentyűk megnyomásával kapcsolhatjuk át
-SOUND: ON állásban a hangjelzés be-, OFF esetén kikapcsolt állapotban van. Az átkapcsolás a CAPS SHIFT és az 1 billentyűk megnyomásával történik.

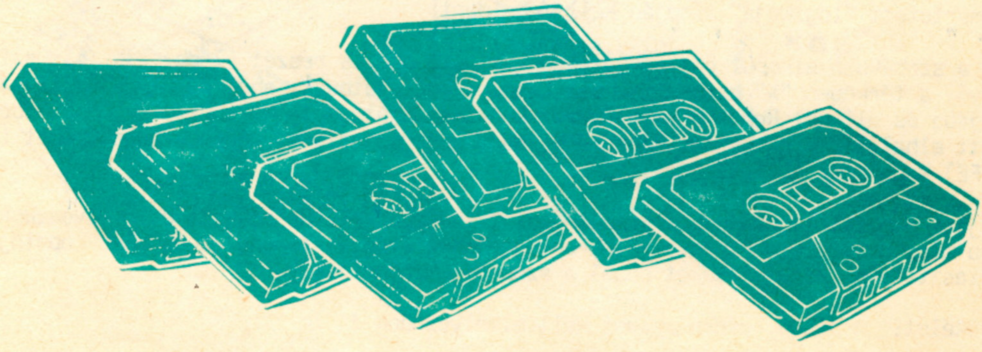
A következő sorban láthatjuk a betöltött file-okra vonatkozó adatokat:

-T: a file típusa ( P : BASIC ; C : gépi kód ; N : számtömb vagy adattömb)
-NAME: a file neve
-LENGTH: a file fejléc szerinti hossza
-ADDR.: töltési cím
-BYTES: a betöltött adatok száma
-E: hiba esetén az aktuális sorban egy csillagot láthatunk. Hibajelzést kapunk, ha:
1. olyan fejlécet olvasunk be, amelyben a LENGTH érték nagyobb, mint 41535
2. a program értelmezhetetlen fejlécet talál
3. a file betöltése után a LENGTH és a BYTES érték nem egyezik (ha a két érték egyezik és mégis csillagot látunk a sor után, a file hibás, visszatöltéskor Tape loading error hibaüzenetet vagy rendszerresetet eredményez. SCREENS esetén például összekeverednek az attributumok)
4. olyan fejléc nélküli file-t töltöttünk be amely hosszabb 41535-nél

Néhány program fejlécében látszólag hibás értékek szerepelnek. Pl. a CAVELON játékprogram főkódjában a LENGTH érték 39000 a betöltött BYTES érték pedig csak 26000. Természetesen hibaüzenetet kapunk, de - mivel másolhatjuk a hibás file-okat is - a program kimentés után működik. Tehát ha ilyennel találkozunk és nem a szalagba töröltünk bele, gondoljunk az ilyen programozói turpisságokra is.

Előfordulhat, hogy a fejléc nélküli file betöltése után "P .......... 0 0 0" jelzést látunk. Ez nem hiba, ha így kimentjük a file-t, hibátlanul működni fog.

A CAPS SHIFT és E billentyűk megnyomása után a másolót "kilőhetjük", de ez csak egy RET assembly-utasításnak felel meg, ezután még célszerű kiadni a következőt: RANDOMIZE USR 0 !



# WHITE LIGHTNING

## OASIS

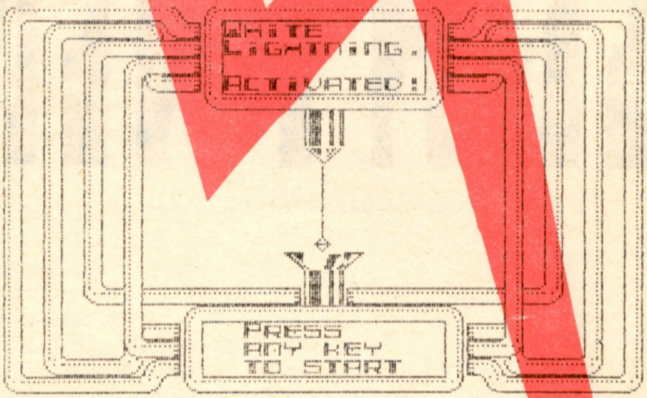
## 1984

# BEVEZETÉS

A ZX Spectrum számítógépekhez forgalmazott felhasználói software-ek jelentős részét a programozást elősegítő, úgynevezett toolkit-programok teszik ki. Ezek általában másolás útján, leírás nélkül kerülnek hazánk Spectrum-tulajdonosainak birtokába. Ez a körülmény célszerű felhasználásukat - enyhén szólva - nehézkessé (pontosabban: lehetetlenné) teszi, különösen akkor, ha például nem csak egy egyszerű rajzolóprogramról, hanem egy olyan bonyolult programozási rendszerről van szó, mint a WHITE LIGHTNING. Az LSI ATSz. Spectrum-programokkal foglalkozó kiadványaiban rendre találkozhattunk már toolkit-programok felhasználói leírásaival, de a WHITE LIGHTNING-ról még nem esett szó. Az alábbiakban egy olyan toolkit-programot mutatunk be részletesen, amely Magyarországon közkézen forog (minden Spectrum-klubban hozzáférhető), de a hozzátartozó leírás szinte sehol sem érhető el. A program ismertetését folytatásokban tervezzük, a későbbiekben megjelenő "SPECTRUM-világ" kiadványok középső négy oldalán. Az elhelyezés nem véletlen: a belső lapok kiszedhetőek és a sorozat befejezése után be lehet köttetni.

Mielőtt még részletesebben foglalkoznánk a WHITE LIGHTNING-gal meg szeretnénk említeni a LIGHTNING-programcsomag "testvéröccsét", a szintén az OASIS Software Inc. által, 1986-ban publikált LASER-családot. Ezen programrendszer LASER BASIC alprogramja rendkívül sok hasonlóságot mutat a WHITE LIGHTNING-gal, nem titkolt továbbfejlesztése. Ezzel a programcsaláddal (LASER BASIC, COMPILER, GENIUS) részletesebben megismerkedhettek a "Sinclair SPECTRUM játék és program III." című, az LSI ATSz. gondozásában 1987-ben megjelent kiadványban. A két programcsalád nyelvezete rendkívül hasonló, aki az egyiket ismeri, annak a másik megtanulása már nem okozhat problémát.

Hát akkor lássuk a medvét...




A WHITE LIGHTNING egy magasszintű programozási rendszer a 48 kbyte-os Spectrum számítógépen. Elsősorban azoknak íródott, akik kedvet éreznek a játékok írásához és ehhez van türelmük megtanulni egy új programozási nyelvet. A LIGHTNING nem egy játéktervező-program és nem olyan, amelylyel egy nap leforgása alatt máris szédületes eredményt érhetnénk el, de lehetőséget nyújt arra, hogy jó programokat írhassunk. Külön előny az is, hogy a WHITE LIGHTNING által megírt programok minden szabadalmi megkötés nélkül értékesíthetőek.

Minden programozóban felvetődik a kérdés: melyik a jó programozási nyelv? Közelítsük meg ezt a kérdést most a játékok oldaláról: BASIC-en kívül Spectrum számítógépeken leggyakrabban használt magasszintű nyelvnek, az assembly-nek három előnye van a többivel szemben: a gyorsaság, a rugalmasság és a tömörség. Ez különösen az arcade-stílusú játékok futtatásakor jelentkezik: a processzor ideje nagy részét a képpont-adatok kezelésével tölti, mégis a játékok legnagyobb része gyors, jól animált, élvezetes. Ebből már adódik a következtetés: ha a programozási szük-

ségletnek megfelelő toolkit-utasítások egy célszerű nyelvben össze vannak foglalva, a program megírási és végrehajtási ideje jelentősen megrövidül (megjegyzendő, hogy elég sok időbe kerül a végrehajtás sebességét optimalizálni).

Hasonlítsuk össze a WHITE LIGHTNING-ot az assemblyvel: ami a nyelv rugalmasságát illeti 300 (!) utasítással rendelkezik amellett, hogy szükség esetén - kombináltan - hozzáférhető a Spectrum-BASIC és a gépi kód is. Tömörségéről annyit, hogy az általa - többek között - kezelt FORTH nyelv legfőbb jellemzője a maximális tömörségre való törekvés. Az assembly nyelv a programozók egyik legkiválóbb eszköze, azonban van néhány hátránya is. Például meg kell tanulni, bár ez mondjuk a többi nyelvre is érvényes. Megtanulása után a programírás kissé körülményes (pl. más nyelvektől eltérően nincs közvetlen védelem a programok kiakadása ellen). Hatásos eredmények eléréséhez igen jól kell ismerni a gépi kódú programozás szabályait. Ebből a szempontból a BASIC jóval előnyösebb: a hibaüzeneteket generáló rutinok kitűnő védelmet nyújtanak a programunknak, az utasítások könnyen elérhetőek és - angolul tudók esetében - rövid idő alatt megtanulhatóak. Ezek a tulajdonságok a hobbiból programozók szemében kiváló nyelvvé emelik a BASIC-et, de a profi játékíróknak a nyelv nem megfelelő mind sebességet, mind rugalmasságot tekintve. Mivel a WHITELIGTNING - részben - a FORTH nyelvre lett alapozva, gyorsasága a gépi kódéhoz hasonló, nem szükséges a gép alapos ismerete és a programok elég jól védve vannak az "elszállás" ellen.

Mielőtt részletesen "elmélyülnénk" a programban nézzük át néhány szó erejéig a WHITE LIGHTNING által nyújtott lehetőségeket:

Sprite-tervezés és UDG-karakterek

A WHITE LIGHTNING tartalmaz egy sprite-generátort és egy 167 ábrából álló készletet, amely sok arcade játék figuráit tartalmazza (Asteroids, Pacman, Scramble, Defender, Space Invaders, City Bomber, Lunar Lander, Frogger, Centipede, Donkey Kong és még sokan mások). Ezenkívül 255 sprite-ot definiálhatunk saját használatunkra, amelyeket forgathatunk, tükrözhetünk, invertálhatunk és még sok más műveletet végezhetünk velük. A munka befejezése után - a későbbi felhasználás céljából - az elkészült sprite-jainkat magnóra menthetjük.

Az IDEAL-nyelv

A LIGHTNING nyelve két részből áll: az egyik a szupergyors Integer-FORTH, amelyet a Fig-FORTH támogat; a másik egy megszakításvezérelt kiterjesztett animációs nyelv az IDEAL (Interrupt Driven Extended Animation Language). A játékíróknak ez a program legfontosabb része: több mint 100 utasítást tartalmaz, amelyeket szabadon keverhetünk mind a FORTH, mind a BASIC-utasításokkal.

Megszakításvezérlés

A FORTH/IDEAL parancsok végrehajthatóak interrupt módban is. Ez azt jelenti, hogy pl. egy játék háttere és előtere egyszerre vezérelhető. Tegyük fel, hogy a játékunk amit írunk, tartalmaz a háttérben egy scroll-t, az előtérben pedig egy sprite-mozgatást. A WHITE LIGHTNING ezeket párhuzamosan végzi el, megszabadítva a felhasználót a komplex időszámításoktól.



OASIS SOFTWARE
9.a Alexandra Parade
Weston Super-Mare
AVON BS23 1QT
ENGLAND

OASIS

Animációs nyelv

A szerzők sok időt fordítottak arra,hogy lehetővé tegyék az egyik gépen LIGHTNING programmal megírt programok futtatását a másik gépen is. Az egyetlen nagyobb változtatás, amire szükség volt, adaptálni a különböző képernyőformátumokat, azaz az oszlopok és a sorok számát. Emiatt már nem kell aggódnunk, ezt elvégzi az IDEAL. Van azonban néhány tiltás, amely nem engedi meg az attributumok mozgatását a pixelek nélkül.

BASIC-lehetőségek

Ha ez idáig még nem ismerkedtünk meg a FORTH nyelvvel, az IDEAL-hoz hozzáférhetünk BASIC-kel is. A megírt programok így veszítenek gyorsaságukból és csillogásukból, de azért még tökéletesek maradnak. A BASIC nagyobb memóriaterületet igényel, ezt jó ha figyelembe vesszük mielőtt még letennénk a FORTH elsajátításáról. A legtöbb hangot és grafikát kezelő utasítást (BEEP,DRAW stb.) belevették a Spectra-Forth nyelvbe, célszerűbb tehát ezeket nem BASIC-ből hívni, mivel FORTH-ban sokkal gyorsabbak.

Kiterjesztések

A FORTH egy kiterjeszthető nyelvjárás, ez volt az oka annak, hogy az IDEAL nyelv mellett bekerült a programba. Az általunk alkalmazni kívánt új szavakat a FORTH és IDEAL szavakkal, vagy a már előzőleg meghatározott szavaink kombinációjával definiálhatjuk.

# I.rész : A SPRITE-GENERÁTOR

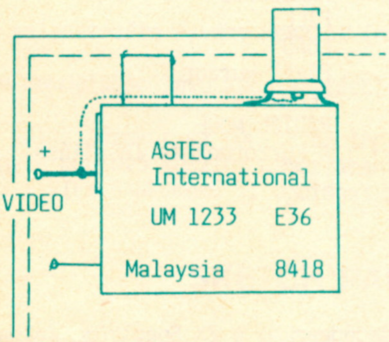
A program ezen része sprite-ok és képernyő-adatok kezelésére szolgáló utasításokat foglal magába, de közvetlenül nem alkalmas sprite-vezérlésre. Ez azt jelenti, hogy egy játékot két fázisban kell elkészítenünk: az egyik fázisban sprite-okat tervezünk és szerkesztünk, a másodikban pedig magát a játékot írjuk meg a LIGHTNING-gal. A gyakorlatban a két fázis tulajdonképpen egymás mellett valósul meg. A program SPT/GEN néven található meg a kazettán, hagyományos módon, LOAD "" paranccsal tölthető be. Önállóan is működőképes, tehát saját - nem a WHITE LIGHTNING-ban történő - felhasználásra is tervezhetünk vele sprite-okat.

Azoknak, akik nem kívánnak új szimbólumokat készíteni, hanem megelégszenek az ismert arcade játékok figuráival, rendelkezésükre áll az arcade-készlet. Ez összesen 167 szimbólumot tartalmaz, megtekintésük a DEMO behívásával lehetséges. Ebben a 'Z' billentyű megnyomásával szelektálhatunk. Az arcade készleten kívül rendelkezésre állnak a demo-sprite-ok is: a szalagon a WHITE LIGHTNING program után találhatjuk őket.

## VIDEO KIMENET A SPECTRUM 16/48K GÉPRE ÉS ZX-81-RE

Mint tudjuk a Spectrumban az ULA állítja elő video-jelet. Ez a hátsó él-csatlakozó egyik lábán is megjelenik, az UHF modulátor ebből állítja elő a 36. csatornának megfelelő antennajelet. A gépre mégsem került külön videocsatlakozó aljzat, pedig gyakran szükségünk lehet rá (pl. videomonitor felhasználásakor vagy a képminőség javításánál). Ha szétszereljük a Spectrumot, hamar beláthatjuk, hogy a video-csatlakozót legcélszerűbb az UHF modulátornál az antennakimenet mellett elhelyezni. A legegyszerűbb az RCA típusú aljzat beépítése, hiszen video-készülékekhez, monitorokhoz ez az egyik leggyakrabban használt típus és hazánkban probléma nélkül beszerezhető. RCA aljzatból is több típus is van, célszerűbb a beépíthető, szerelhető kivitelt megvenni, mert az átjátszó vezetékre szerelhető lengő RCA aljzat beszerelésekor gondokba ütközhetünk. Az ASTEC-típusú UHF modulátor bal oldalán egy műanyag tokot találunk, amelynek az antennakimenettől távolabb eső bemenetére csatlakozik egy vezeték. Itt kapja a modulátor a video-jelet.

a/ Egyszerű megoldás, túlvezérelt jellel:

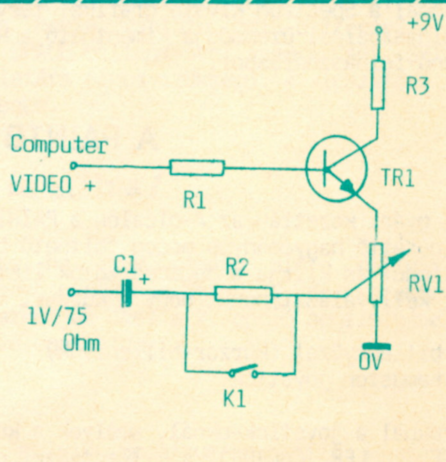



Az ábrán látható módon forrasszuk fel az RCA aljzatot az UHF modulátor oldalára, majd az RCA aljzat középső pólusát egy hajlékony vezetékszakasz segítségével kössük össze a video-lábbal. A Spectrum műanyag dobozának felső részén az RCA aljzatnak megfelelő méretű kivágást kell eszközölnünk (nem árt, ha esztétikai szempontokat is figyelembe veszünk). A megoldás működőképes, de a szabvány 1 V/75 Ohm video-jel helyett 3 V-os jelet ad le, tehát ha a vevőkészülék nem a Spectrumhoz lett beállítva, a túlvezérelt video jel torzítást okozhat.

b/ A túlvezérelt jel beszabályozása

E megoldás, hogy a video-jelet egy egytranzisztoros emitterkövető kapcsoláson vezetjük át. Így megkapjuk az 1 V/75 Ohm értéket. Az alkatrészlista:

| | | | |
|---|---|---|---|
| Tranzisztor | 1 db. | 2N 2219 | TR1 |
| Kondenzátor | 1 db. | 100 µF 16V ELKO | C1 |
| Ellenállások | 1 db. | Trimmer 100 Ohm | RV1 |
| | 1 db. | 1 kOhm 1/4 W | R1 |
| | 2 db. | 68 Ohm 1/4 W | R2,R3 |
| Kapcsoló | 1 db. | kétállású | K1 |




Az áramkör szíve a 2N 2219 típusú n-p-n szilícium tranzisztor. A szükséges 1 V értéket a trimmer ellenálláson kell beszabályozni. Fontos, hogy a trimmerellenállás szénréteg ellenállás legyen, mert tekercselt huzal esetén a szórt induktivitás a magas frekvencián elhangolódáshoz vezethet.

Spectrum esetén a "K" kapcsolónak zárt állásban kell lennie. A kapcsolás a ZX-81-en is működik, de ez esetben a "K" kapcsolót nyitnunk kell.

# PAINTBOX III/1.

PRINT 'N' PLOTTER

PAINTBOX

GRAPHICS TOOLKIT

## A VEZÉRLŐBILLENTYŰK FUNKCIÓI

| | | | |
|---|---|---|---|
| '5': | Kurzor balra | '6': | Kurzor fel |
| '7': | Kurzor le | '8': | Kurzor jobbra |
| '9': | GRAPHICS | '0': | Kiválasztás |
| 'A': | ARC | 'BREAK: | Kiszállás |
| 'E': | ERASE | 'F': | FILL |
| 'H': | CIRCLE | 'J': | LOAD |
| 'K': | Kereszt alakú kurzor | 'M': | Pixel-kurzor |
| 'N': | OVER | 'P': | PERMANENT |
| 'Q': | PLOT | 'R': | RADIAL MODE |
| 'S': | SAVE | 'SHIFT': | A kurzormozgás lassítása |

## Bevezetés

A PAINTBOX egy önálló grafikus programozási segédeszköz a ZX Spectrum felhasználóinak. A program írói egyetlen átfogó gépi kódú rutinba foglalták össze azokat az eszközöket, amelyek a számítógéppel való rajzoláshoz kiváló segítséget nyújtanak.
A PAINTBOX-ot úgy írták meg, hogy teljesen kompatibilis legyen a JOTTER nevű grafikus tervező programmal, vagyis azzal együtt is lehessen használni. Ezzel a két programmal a Spectrum kiváló grafikai tervező számítógéppé válik. A program valamennyi felhasználási üzemmódja kimenthető a későbbi BASIC vagy gépi kódú programokban történő felhasználáshoz.

## A PAINTBOX behivása

Az gyári kazettán az A-oldalon a PAINTBOX, a B-oldalon pedig a DEMO található, mindkettő hagyományos módon tölthető be. Betöltődés után a program automatikusan elindul és kiírja a segédeszközök listáját. A kívánt segédeszköz meghatározása után ki kell választanunk, hogy a kurzort milyen módon kívánjuk irányítani:

- billentyűzet (kurzor-billentyűk)
- Kempston joystick

Azoknál a joystick-oknál, amelyek a kurzor-billentyűkre adják a vezérlést (PROTEK, AGF, FULLER, CURSOR), a billentyűzet opciót kell választanunk.

## Az UDG Editor

Mint ismeretes, a Spectrum egyszerre 21 db UDG karakter tervezésére ad lehetőséget, de nemritkán ennél többre is szükségünk lenne. A PAINTBOX-ban egyidőben összesen 84 UDG-karaktert használhatunk fel. Ez nem befolyásolja a normál karakterkészletet és a kibővített grafikus készletet beépíthetővé teszi bármilyen saját programba. Az UDG Editor választása után azonnal egy újabb menüt láthatunk, amelyben a következő opciók állnak rendelkezésünkre:

- VIEW UDG BANKS (az UDG készletek szemléje)

Az opció választásakor megjelenik a négy rendelkezésre álló UDG-készlet. Maga az UDG terület az A-U közötti betűket tartalmazza és legfelül helyezkedik el. Alapállapotban a készletek a program írói által gyártott karakterekkel vannak feltöltve de ezek természetesen tetszés szerint átdefiniálhatóak.

- RECALL n (UDG-készlet behívása)

Az opció választására kívánt készlet (n) betöltődik az UDG-területre.

- SAVE (a memóriában lévő készlet tárolása)

Az opció választásakor az UDG-területen lévő felhasználói karakterkészlet átmásolódik az egyik meglévő készlet memóriában fenntartott helyére. A szalagra történő kimentéshez a VIEW BANK-menü 5. opcióját kell választanunk.

## SKETCH PAD / vázlatfüzet /

A VIEW BANKS-menüből a 4. opció (MAIN MENU) választásával térhetünk vissza a főmenübe. Itt a SKETCH PAD opciót választva előhívhatjuk a vázlatkészítő füzetet, amelyen ki lehet próbálni az egymás mellé szánt UDG-karaktereket. A képernyő tetején a pillanatnyilag aktuális karakterkészlet látható.
A villogó kurzort bármelyik irányban végigvihetjük a készlet mentén. A kiírásra szánt karakterek kiválasztásához nyomjuk meg a hozzá tartozó betű-billentyűt. A kiíratni nem kívánt karaktereket felülírhatjuk SPACE billentyűvel (sem a CAPS SHIFT billentyű, sem pedig a GRAPHICS üzemmód használatára nincs szükség). A visszatérés a főmenühöz a '4' billentyű megnyomásával lehetséges.

## DRAWING BOARD / rajztábla /

Bármilyen grafikus segédeszköz haszontalan lenne, ha csak néhány előre definiált karakterkészletet tartalmazna. A PAINTBOX programban azonban mind a 84 karaktert tetszés szerint módosíthatjuk illetve újratervezhetjük őket a 2. opció segítségével, ahol a következő opciókat tartalmazó listában válogathatunk.

-NORMAL CHARACTER

Az opció választásakor a gép megkérdezi, hogy melyik Spectrum-karaktert kívánjuk módosítani. Választásunkat a megfelelő billentyű (szükség esetén a CAPS SHIFT ) megnyomásával jelezhetjük a gépnek

-UDG CHARACTER

Az opció választásakor a gép megkérdezi, hogy melyik UDG-karaktert kívánjuk módosítani a képernyő tetején látható készletből. Választásunkat a karakterhez tartozó betű (A-U) megnyomásával közöljük a géppel

-BLANK CHARACTER

Az opció választásakor egy üres karaktert láthatunk. Az elméletileg 8x8 pixel méretű karakter bal felső sarkában megjelenik egy vörös kurzor, amelyet a karakter szimbólumán belül az irányító billentyűkkel mozgathatunk. Ha valamelyik pixel-pozícióban megnyomjuk a '0' gombot, az a karakterben INK színű lesz. A rács mellett látható az adott pixelsort a memóriában definiáló decimális byte is. Ha a karakter definiálását befejeztük, nyomjuk meg az 'ENTER' billentyűt. Ekkor megjelenik egy menü, amelyben a következő opciók között válogathatunk:

-FILEWORK: elhelyezi a karaktert az általunk választott készletben
-INVERSE : invertálja a karaktert
-ROTATE : elforgatja a karaktert 90 fokkal
-MIRROR : a karakter tükörképét adja

## Szalagműveletek

Az UDG-menü 5.(SAVE) opciója lehetővé teszi a memóriában lévő négy karakterkészlet egyidejű kimentését. Az opció választása után be kell gépelnünk a karakterkészlet max. 6 karakter hosszúságú nevét, majd a gép ezen a néven szalagra menti az UDG-ket a kezelésükhöz szükséges kódrutinnal együtt. Az UDG-menü 4.(LOAD) opciója az 5. opció szerint kimentett készleteket tölti be a magnóról.

## UDG készletek

Ha az UDG-készleteket nem a PRECISION PLOTTER (ld. később) képernyőinél kívánjuk felhasználni, valamely UDG-karakterállományt tartalmazó BASIC-program betöltésekor - az UDG-ket megvédendő - a következő parancsot kell kiadnunk:

```
CLEAR 64567: LOAD "": LOAD "(az állomány neve)" CODE
```

Az UDG-készletekből négy darabot tudunk egyszerre tárolni, de ahhoz, hogy bármelyik aktuális készletté váljon, saját kódrutin szükséges. Ha - más programban történő felhasználás esetén - valamelyik karakterkészlet valamelyik karakterét a PRINT utasítás használatával ki kívánjuk a képernyőre íratni, a PRINT előtt lévő valamelyik sorban meg kell hívnunk a készlethez tartozó rutint. A különböző készletekhez tartozó rutinok indítása a következő módon történik:

```
1.készlet: RANDOMIZE USR 65153
2.készlet: RANDOMIZE USR 64958
3.készlet: RANDOMIZE USR 64763
4.készlet: RANDOMIZE USR 64568
```

Fontos megjegyzés: Ha a programunkban alkalmazni kívánunk RND függvény által előállított véletlenszerű számokat, előfordulhat, hogy a gépi kódot hívó RANDOMIZE utasítás megváltoztatja a számainkat. Célszerű ilyenkor egy változót használni a karakterkészletek aktivizálására:

```
LET a= USR (a kívánt készlet címe): RANDOMIZE a
```



## Egyszerű BASIC tool-kit lehetőségek

a/ DELETE

Ha BASIC programunk egy hosszabb részétől akarunk megszabadulni, és nincs kéznél DELETE segédrutin, akkor segíthet ez az egyszerű POKE-PEEK trükk. Hátránya, hogy csak a program végén lévő blokkokat képes törölni. Törli a változókat is. A két POKE-nak és a CLEAR-nek egy sorban kell lennie és egyikük sem hagyható el. Minden esetben az ezen utasításokat követő sortól a program végéig törlődik a BASIC. A mintaprogram futtatása után a 70-90. sorok eltűnnek.

```
10 REM
20 REM
30 REM
40 POKE 23627,PEEK 23637:
   POKE 23628,PEEK 23638:
   CLEAR
70 REM
80 REM
90 REM
```

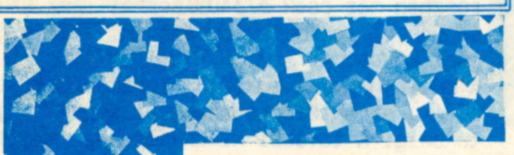
b/ RENUMBER

Gyakran szükségünk lehet BASIC programunk írása közben a programsorok átsorszámozására. Nem mindig van kéznél gépi kódú RENUMBER, nem beszélve arról, hogy többségük komplex program része, s lehet hogy a memóriatakarékosság miatt nincs lehetőség a TOOL-KIT-et betölteni. Az itt közölt program a RUN 9986 utasítással bekéri az átsorszámozni kívánt blokk startsorának számát, majd az átsorszámozás lépésközét. Ezután a teljes BASIC programot átsorszámozza a 9985. sorig bezárólag.

```
  10 REM
  14 REM
  16 REM
 100 REM
 123 REM
9986 DEF FN p()=PEEK 23635+256*P
EEK 23636
9987 DEF FN l(a)=256*PEEK a+PEEK
 (a+1)
9988 LET p=FN p()
9989 INPUT "start sorszam: ";so
9990 INPUT "lepeskoz: ";io
9991 PRINT "regi";TAB (10);"uj"
9992 LET l=FN l(p)
9993 IF l>9985 THEN  STOP
9994 PRINT l;TAB (10);so
9995 POKE p,INT (so/256)
9996 POKE p+1,so-INT (so/256)*25
6
9997 LET so=so+io
9998 LET p=p+4+PEEK (p+2)+256*PE
EK (p+3)
9999 GO TO 9992
```

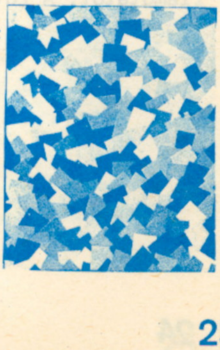
## Modern művészet BASIC-ben

Bizonyára kevesen mondanák meg erről a néhány soros BASIC programról, valójában mit is csinál. Igen látványos grafikai montázst jelenít meg a képernyőn, az elemek keveredésével két azonos kép megjelenésének valószínűsége igen csekély.
Ha a 20. sor végén lévő számot megváltoztatjuk, pl.1,2,3,4, stb. értékre, különböző hatásokat érhetünk el, az egyszerű vonalkomponensektől egészen Victor Vasarely művészi szintjét megközelítő grafikáig.





```
 10 BORDER 7: PAPER 7: INK 0: C
LS
 20 LET a=INT (RND*70)+100
 30 LET b=INT (RND*60)
 40 LET c=INT (RND*3)-4
 50 FOR i=a TO b STEP c
 60 PLOT i,i: DRAW 0,175-2*i
 70 DRAW 255-2*i,0
 80 DRAW 0,-175+2*i
 90 DRAW -255+2*i,0
100 INK RND*6: BORDER RND*6
110 OVER 1
120 NEXT i
130 GO TO 20
```



## A RAJZFILMKÉSZÍTÉS kezdeti lépései ZX Spectrumon

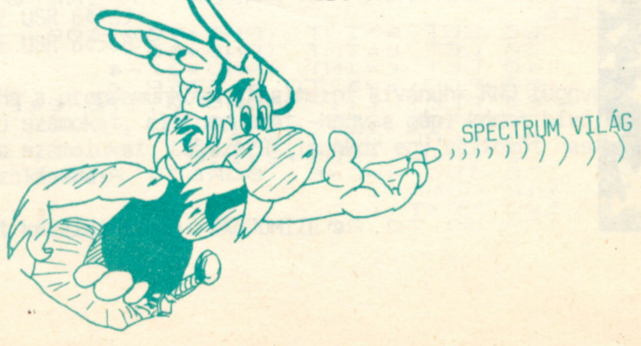
Az animáció a rajzfilmkészítéshez hasonlóan több kép egymást követő előhívásával oldható meg. Célszerű a képeket (fázisokat) előre elkészíteni, hiszen az animáció közben ezt nem tudjuk megfelelő gyorsasággal elvégezni. Az alábbiakban ismertetett gépi kódú rutin két főbb részből áll. Az első rész (40000-40068) segítségével hívhatjuk elő az egyes fázisokat a képernyő előre meghatározott részére, míg a második rész (40069-40117) az általunk elkészített fázisokat helyezi el a memória hátsó területére. Egy képernyő-ablak (fázis) mérete nem kötött, bár az itt ismertetett rutin felhasználásával max. 32 karakter széles és 8 karakter magas ablakot használhatunk. Nem történik meg a képernyőharmadok közötti címvizsgálat sem (ez egyébként a box-scroll-oknál ismert módon módosítható), ezért vigyázzunk, hogy az ablakunk ne csússzon át a képernyőharmadokon!

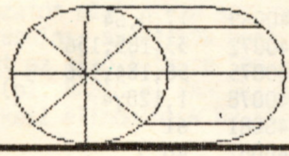
Mintaprogramunkban 4 db 16 karakter széles és 8 karakter magas méretű animációs fázist készítünk (egyébként a fázisok száma tetszőleges, határt csak a rendelkezésre álló memória szab). Függetlenül a fázis megjelenítési pozíciójától, a fázisok elkészítését célszerű egy bizonyos helyen elvégezni, legyen ez pl. a képernyő bal felső sarka. Ennek megfelelően a DE regiszterbe 16384-et állítunk be. A HL regiszterben megadott 40121. cím a memória azon címe, ahonnan a fázisok tárolása kezdődik. Az A regiszterben adjuk meg az aktuális animációs fázis számát (1,2,..), BC-ben pedig az animációs ablak adatainak végösszegét. Ez jelen esetben 16x8-as ablak révén 1152, vagyis a képpont adatok mellett az ATTR adatokat is tartalmazza. Ez az érték azért szükséges, mert fázisonként ennyivel léptetjük tovább a memóriamutatót. A rutin ezt követően elvégzi a képpont adatok, majd az attributumok átmásolását a megfelelő helyre. Az adatok visszahívása fordítottan történik, de itt előre beállíthatjuk, a megjelenítés pozícióját a 40118/40119 címeken. Mindkét mechanizmusban tetszés szerint módosítható az ablak mérete a meghatározott címeken.

A gépi kódú rutin vezérlése BASIC-ből történik. Görgessünk egy hordót a képernyő felső harmadában. 4 fázist készítünk. A BASIC program első részében a képernyőfényképezés történik, vagyis egymás után felrajzoljuk a fázisokat, beállítjuk a fázis számát, majd elvégezzük a tárolást (a 9000. sorban elhelyezett szubrutin segítségével). A 4 fázis "fényképezése" után következik a vezérlő szakasz, melynek végrehajtója a 9100-as sorban elhelyezett szubrutin. Itt állítjuk be a fázis számát ill. a megjelenítési pozíció kezdőcímét. Fokozatosan csökkenő értékű időtagot beiktatva a hordó forgása egyre gyorsabbá válik. A soron következő fázis képe az előzőét törli, egy karakteroszlop kivételével, ezért hagyunk az ablak két szélén 1-1 üres karakteroszlopot.

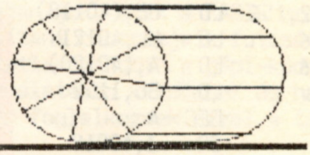
Ha a képeket már egyszer eltároltuk a memóriában, azt kód formájában kimenthetjük a rutinnal együtt, így legközelebb már nem kell a rajzolási műveletet végrehajtani. Jelen esetben az eltárolt négy fázis kód hossza 4x1152=4608 byte, a rutin és a hivatkozási címek együttes hossza pedig 121 byte, így a kimentés a SAVE "név" CODE 40000,4729 utasítással végezhető el. Természetesen ezután a BASIC részben már csak a 340. sortól lesz szükségünk a programra, de ezt is átírhatjuk gépi kódba.

A rutin nagyon sokrétű lehetőséget tár elénk, az oktatóprogramoktól a saját játékokig. A további kimunkáláshoz sok sikert !



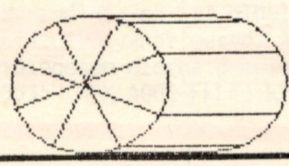


1. Fázis

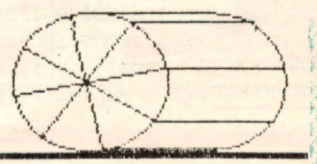


2. Fázis

```
 10 REM elso fazis
 20 GO SUB 600
 30 PLOT 10,144: DRAW 108,0
 40 PLOT 40,112: DRAW 0,61
 50 PLOT 19,165: DRAW 44,-44: D
RAW 47,0
 60 PLOT 19,121: DRAW 44,44: DR
AW 47,0
 70 LET a=1: GO SUB 9000
 80 PAUSE 100: CLS
 90 REM masodik fazis
 100 GO SUB 600
 110 PLOT 23,167: DRAW 35,-51: D
RAW 47,0
 120 PLOT 34,113: DRAW 13,60: DR
AW 47,0
 130 PLOT 15,126: DRAW 52,34: DR
AW 47,0
 140 PLOT 11,149: DRAW 60,-12: D
RAW 47,0
 150 LET a=2: GO SUB 9000
 160 PAUSE 100: CLS
 170 REM harmadik fazis
 180 GO SUB 600
 190 PLOT 28,171: DRAW 25,-56: D
RAW 47,0
 200 PLOT 12,154: DRAW 57,-25: D
RAW 47,0
 210 PLOT 12,132: DRAW 57,25: DR
AW 47,0
 220 PLOT 27,115: DRAW 25,56: DR
AW 47,0
 230 LET a=3: GO SUB 9000
 240 PAUSE 100: CLS
 250 REM negyedik fazis
 260 GO SUB 600
 270 PLOT 33,173: DRAW 13,-60: D
RAW 47,0
 280 PLOT 15,159: DRAW 52,-34: D
RAW 47,0
 290 PLOT 11,137: DRAW 60,12: DR
AW 47,0
 300 PLOT 22,118: DRAW 36,52: DR
AW 47,0
 310 LET a=4: GO SUB 9000
 320 PAUSE 100: CLS
 330 PLOT 0,111: DRAW 255,0: PLO
T 0,110: DRAW 255,0: PLOT 0,109:
DRAW 255,0
 340 REM vezerlo blokk
 350 LET a=1: LET b=0: LET c=0
 360 LET i=7
 370 GO SUB 9100
 380 LET a=a+1
 390 IF a>4 THEN LET a=1
 400 LET c=c+1
 410 PAUSE i
 420 IF c<16 THEN GO TO 360
 430 GO SUB 9100
 440 LET a=a-1
 450 IF a<1 THEN LET a=4
 460 LET c=c-1
 470 PAUSE i
 480 IF c>0 THEN GO TO 430
 490 LET i=i-1
 500 IF i<1 THEN STOP
 510 GO TO 370
 600 CIRCLE 41,143,31: PLOT 88,1
74: DRAW 0,-62,-PI: PLOT 40,174:
DRAW 48,0: PLOT 40,112: DRAW 48,
0: RETURN
9000 POKE 40120,a: RANDOMIZE USR
40069: RETURN
9100 POKE 40120,a: POKE 40118,c:
POKE 40119,64: RANDOMIZE USR 400
00: RETURN
```

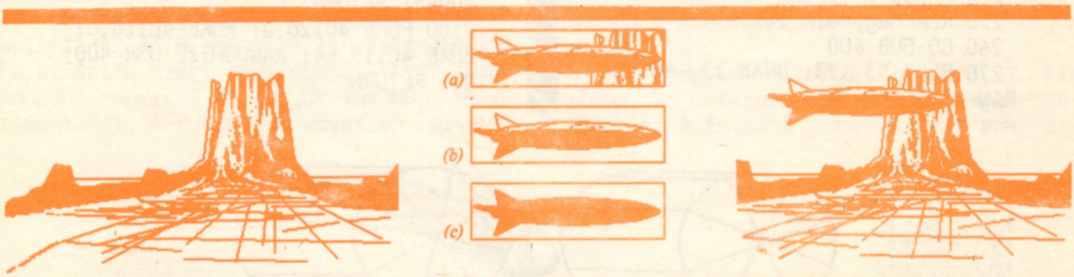


3. Fázis



4. Fázis

```
40000   237,91,182,156  LD    DE,(40118)
40004   33,185,156      LD    HL,40121
40007   58,184,156      LD    A,(40120)
40010   1,128,4         LD    BC,1152
40013   61              DEC   A
40014   40,3            JR    Z,40019
40016   9               ADD   HL,BC
40017   24,250          JR    40013
40019   62,64           LD    A,64
40021   1,16,0          LD    BC,16
40024   197             PUSH  BC
40025   237,176         LDIR
40027   193             POP   BC
40028   235             EX    DE,HL
40029   9               ADD   HL,BC
40030   235             EX    DE,HL
40031   61              DEC   A
40032   32,246          JR    NZ,40024
40034   22,88           LD    D,88
40036   58,182,156      LD    A,(40118)
40039   95              LD    E,A
40040   58,183,156      LD    A,(40119)
40043   214,64          SUB   64
40045   203,63          SRL   A
40047   203,63          SRL   A
40049   203,63          SRL   A
40051   130             ADD   A,D
40052   87              LD    D,A
40053   62,8            LD    A,8
40055   1,16,0          LD    BC,16
40058   197             PUSH  BC
40059   237,176         LDIR
40061   193             POP   BC
40062   235             EX    DE,HL
40063   9               ADD   HL,BC
40064   235             EX    DE,HL
40065   61              DEC   A
40066   32,246          JR    NZ,40058
40068   201             RET
40069   17,0,64         LD    DE,16384
40072   33,185,156      LD    HL,40121
40075   58,184,156      LD    A,(40120)
40078   1,128,4         LD    BC,1152
40081   61              DEC   A
40082   40,3            JR    Z,40087
40084   9               ADD   HL,BC
40085   24,250          JR    40081
40087   235             EX    DE,HL
40088   62,64           LD    A,64
40090   1,16,0          LD    BC,16
40093   197             PUSH  BC
40094   237,176         LDIR
40096   193             POP   BC
40097   9               ADD   HL,BC
40098   61              DEC   A
40099   32,248          JR    NZ,40093
40101   33,0,88         LD    HL,22528
40104   62,8            LD    A,8
40106   1.16,0          LD    BC,16
40109   197             PUSH  BC
40110   237,176         LDIR
40112   193             POP   BC
40113   9               ADD   HL,BC
40114   61              DEC   A
40115   32,248          JR    NZ,40109
40117   201             RET
-----------------------------------------
40118   0               DATAB 0
40119   64              DATAB 64
40120   2               DATAB 2
40121   ...             Kép adatok
                        startcíme
```



# GÉPI KÓD TANFOLYAM I.

Az itt induló sorozatot elsősorban azoknak ajánljuk, akik a BASIC programozás korlátait szeretnék kikerülni, szeretnék meggyorsítani programjaik futását, egyszóval meg akarják tanulni ezt az első látásra "bevehetetlen erődöt", a gépi-kódot. Sajnos a magasszintű nyelvek (pl. BASIC v. PASCAL) használatában kezdőknek és haladóknak egyaránt az alapoktól kell elindulniuk ha először foglalkoznak ezzel a témával.

Mi is az a gépi kódú program ?

A gépi kódú program számok meghatározott sorozata. Ezek a számok (a továbbiakban kódok) a számítógép memóriájában találhatók, innen olvassa ki ezeket a mikroprocesszor a gép központi egysége. A kódok egyben gépi kódú utasítások is. A processzor a memóriából beolvasott kódot azonosítja, és egy végrehajtási láncot párosít hozzá, ugyanis egy kis belső könyvtárral rendelkezik, és az ott található mikroprogramokat futtatja. Ha a memóriában pl. 201 lett elhelyezve, és ezt értelmezi a processzor, akkor neki erre egy visszatérést kell végrehajtania (részletesebben ld. később). Ez így mind szép, de gépi kódban programot írni csak a számokkal kicsit nehézkes lenne, még gyakorlottabbak sem biztos hogy kapásból választ tudnának adni arra, mit jelent a 197 vagy a 73. Ezért vezették be az assembly formátumot. Megoszlanak a vélemények afelől, hogy az assembly nyelv gépi kód-e, vagy valami más ? Nos az assembly a gépi kód jelentéstartalmát tükröző, azt az ember számára érthetőbbé tevő nyelvezet. Az assembly nyelvben minden gépi kódhoz (számhoz) egy megfelelő utasítás (mnemonik) tartozik. Ez az angol kifejezések rövidítéseiből kialakult kódrendszer. A 201-nek pl. RET az assembly megfelelője, így már mindjárt emberközelibb (RET=RETURN vagyis visszatérés). Az assembly nyelvezet megkönnyíti a felhasználó számára a gépi kódú programok megírását. A gép viszont az assembly nyelvezetet közvetlenül nem érti meg, tehát azt egy megfelelő fordítóprogram segítségével gépi kóddá kell alakítanunk. Ezt a fordítóprogramot hívjuk assemblernek. A disassembler pedig ennek a fordítottja, vagyis a nehezebben értelmezhető gépi kódot fordítja át nekünk felhasználóknak assembly nyelvezetre.

Egységes assembly nyelv nem létezik, csakúgy mint egységes BASIC sem, bár itt az eltérések jóval nagyobbak. A gépi kódú programot minden számítógépben a processzor (CPU=Central Processing Unit=központi vezérlő egység) futtatja, így nem is csoda, hogy pl. a Commodore 64 assembly nyelvezete más, mint a ZX Spectrum-é, hiszen a Commodore-ban a processzor típusa más, 6502-es és nem Z-80-as, mint a Spectrumban. Bár felvetődhet a kérdés, miért nem lehet lefuttatni a Spectrumon minden gond nélkül más - Z-80 processzorral működő - számítógép (pl. VIDEOTON TV Computer) gépi kódját. Egészen egyszerűen azért, mert a memória felosztása - beleértve a képernyőmemóriát is - teljesen más, nem beszélve a ROM program eltéréseiről.

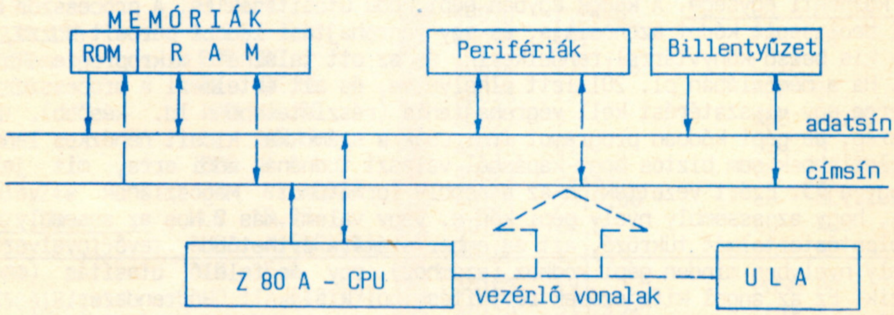
Ha gépi kódban programozunk, feltétlenül ismernünk kell gépünk memóriatérképét, mert a gépi kódú programot nem mindenhol tudjuk a memóriában elhelyezni ill. lefuttatni. Minden gép memóriatérképe megtalálható a géphez mellékelt felhasználói kézikönyvben, így a Spectrumé is. Most itt csak az összefüggések értelmezéséhez szükséges információkat közöljük.

A gép ROM memóriája (Read Only Memory=csak olvasható memória) a memóriaterület elején található, a 00000-16383 (HEX 0000-3FFF) címek között. Ezt a területet nem tudjuk saját célra kihasználni, az itt elhelyezett értékeket nem tudjuk felülírni, az a bekapcsolás pillanatától ugyanaz marad. A további memóriahelyek már a RAM (Random Access Memory=tetszőlegesen elérhető memória) területéhez tartoznak, vagyis 48K-s gép esetén 16384-65535 (HEX 4000-FFFF) között találjuk a RAM memóriát.

# GÉPI KÓD TANFOLYAM I.

Feltehetjük a kérdést, mi az oka annak, hogy a memória éppen 65535-ig tart és nem más értékig pl. 75000-ig. A dolog végtelenül egyszerű: a processzor egy negyvenlábú IC, mely lábakból ha csak hármat (a tápfeszültséget, a földet és az órajelet) bekötünk, már működőképes, de értékes munkát még nem végez. A processzor csak akkor válik hasznos egységgé, ha a számára érthető gépi kódot futtatni is tudja. A gépi kódú program pedig nem a processzorban kerül eltárolásra, hanem a memóriában (memória IC-kben). Ebből azt a következtetést vonhatjuk le, hogy a processzornak a memória IC-khez valahogyan hozzá kell férnie.

A további magyarázathoz tekintsük meg az elvi működés blokkdiagramját:




Az adatáramlás folyamatos biztosításához két fő kommunikációs csatorna található a számítógépben. Az egyik a címsín, a másik pedig az adatsín. A címsín 16, míg az adatsín 8 vonalból áll. A címsín arra jó, hogy a processzor kiválaszthassa rajta keresztül azt a memóriacímet vagy perifériacímet, amellyel a továbbiakban kommunikálni akar (pl. adatot betölteni vagy kivinni). Az adatsínt használja fel a processzor az adat továbbítására. Az adatáramlás persze önmagától nem indul meg, ezt a processzor ún. vezérlő vonalakon keresztül befolyásolja. 13 vezérlő vonalat használ (erre még kitérünk a későbbiekben).

Vizsgáljuk meg egy kicsit a bitek értelmezését. A számítógép kettes számrendszerben dolgozik, így memóriacelláiban csak egyes vagy zérus értékek lehetnek. Tévedés ne essen, a memóriacellát nem szabad összekeverni a memóriacímmel! Egy memóriacímen ugyanis 8 cella található, tehát 1 címen 8 bitet lehet tárolni, azaz 1 byte-ot (1 címet). El ne felejtsük egy byte = nyolc bit.

Egy byte bit-térképe a decimális konverzió szerint a következő:

| Bit-ek sorszáma | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Értéke | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |

Ez azt jelenti, hogy ha az adott byte-ban a 6. a 3. és az 1. bit-ek értéke egyes (01001010) akkor az adott bit-eknek megfelelő decimális értékeket összegezni kell (64+8+2) és így alakul ki az összérték (74). Ha minden értéket összegzünk, vagyis az összes bit tartalma egyes, akkor végeredménynek 255-öt kapunk. Ez az a maximális érték amit egy memóriacímen el tudunk helyezni.

A címsínen - mint említettük - 16 különböző címvonal van és ez két 8 bites fővonalra osztható. Kezelhetőségük mint két különálló de egybetartozó byte. Az egyiket 'felső' a másikat 'alsó' byte-nak is szokták nevezni. Ha az alsó byte értéke meghaladja a 255-öt, pl. 256 akkor az alsó byte lenullázódik és a felső byte értéke eggyel nő. Így abban az esetben, ha mindkét byte összes bitje egyes akkor 255*255 lesz a végeredmény, ez pedig pontosan 65535. Nos, ezért tudunk összesen ennyi memóriahelyet megcímezni.

# GÉPI KÓD TANFOLYAM I.



MINDEN MEMÓRIAHELYHEZ TARTOZIK EGY CÍM 00000-65535-IG.

MINDEN EGYES CÍM ÉRTÉKE (tartalma) 0-255-IG TERJEDHET.

Jogosan vetődhet fel a kérdés, vajon hogyan lehetséges az, hogy a 80 K-ra kibővített vagy a 128-K-s SPECTRUM - melyek szintén Z80 processzorral működnek - a kibővített memóriát is használni tudják?

Nos igaz, a Z80 egyidőben csak 65535 címet "lát", de mi egy adott címhez több memóriahelyet is hozzárendelhetünk, ezek pedig különböző ún. lapokon találhatók. A számítógép egy megfelelő lapozó mechanizmus segítségével a program futása közben át tud kapcsolni az egyes lapok között, így előfordulhat, hogy pl. az 50132-es cím tartalma az 1-es lapon zérus, de a 2-es lapon ettől eltérő érték. (A lapozásról később majd részletesebben lesz szó!)

Ahogy az előzőekben megismertük a gépi kód közvetlen bevitelére BASIC-ből a legkézenfekvőbb - de hosszabb programok esetén időigényes - megoldás a POKE utasítás használata. Próbáljuk most begépelni a következőt: POKE 1365,35

A POKE utasítással a RAM memória megfelelő helyét tölthetjük fel egy adott értékkel. Jelen esetben az 1365-ös ROM címre szeretnénk 35-öt beírni. Tudjuk, hogy a ROM terület nem írható felül, de erről szeretnénk is meggyőződni. A memória visszaolvasására BASIC-ből a PEEK függvény alkalmas, amelyet a PRINT utasítással együtt szoktunk felhasználni. Gépeljük tehát be: PRINT PEEK 1365

Ha megnyomjuk az ENTER-t, az eredmény nem marad el: nem a 35-öt kapjuk vissza, hanem a 201 jelenik meg a képernyőn.

A ROM terület írás-védelem alatt áll, vagyis ezen a területen olyan értékek lettek elhelyezve, amelyeket csak olvashat a mikroprocesszor, de ide írni nem tudunk. Ezeknek a memóriahelyeknek a tartalmát a számítógép még a kikapcsolás után sem felejti el. Fontos ez a terület, hiszen ez teszi lehetővé számunkra, hogy a gép megértse a BASIC nyelven begépelt programjainkat, még konkrétabban operációs rendszert biztosít amely lehetővé teszi, hogy egyáltalán tudjunk a géppel kommunikálni (kurzor, megfelelő szerkesztő stb.). A gép bekapcsoláskor az itt elhelyezett ROM programot automatikusan futtatni kezdi. Az operációs rendszer a BASIC nyelvet értelmező és gépi kóddá alakító fordítóprogramot, az interpreter-t egy szubrutinként kezeli, melyet ha szükséges aktivizál. Természetesen az interpreter ezen belül további szubrutinokat tartalmaz. Megállapíthatjuk, hogy a BASIC program futtatásakor (vagyis ha az operációs rendszer a RUN, GO TO vagy GO SUB utasítással találkozik) az interpreter sorról sorra lefordítja az adott BASIC sort gépi kódra, és a mikroprocesszor ezután futtatja. A BASIC ezért lassú, hiszen a fordítási idő sokat jelent.

Itt még tisztáznunk kell egy másik idegen fogalmat is, mi az a compiler ?

Ez is egy fordítóprogram, de nem szabad összekeverni az interpreterrel. A compiler-t külön töltjük be a szabad RAM területre. A compiler segítségével a memóriában levő teljes BASIC program egyidőben átfordítható gépi kódú programmá, és a fordítás után a lefordított változatot külön futtathatjuk, mint egy gépi kódú programot. Ezzel igen nagy sebességgyorsítást érhetünk el, de még mindig nem érhetjük be a tisztán gépi kódú program sebességét. Mindezek tisztázása után próbáljuk meg mégis a gépi kódú programozás rejtelmeit megismerni.

Az előbb egy ROM címre próbáltunk értéket elhelyezni - ez nem sikerült -. Most próbálkozzunk meg egy RAM címmel:

```
POKE 50120,35
```

majd

```
PRINT PEEK 50120 és ENTER
```

után a 35 megjelenik a képernyőn.

# GÈPI KÒD TANFOLYAM I.

Ezt a műveletet decimális számokkal hajtottuk végre. Tudjuk, hogy a számítógép csak bináris (kettes számrendszerbeli) számokat tud kezelni. Mégis a gépi kódú listák nagy részében találkozunk a hexadecimális (16-os számrendszerbeli) számokkal. Miért jó ez a számrendszer?

Megoszlanak a vélemények a hexadecimális számok használatával kapcsolatban. A jól bevált decimális (10-es) számrendszer emberközelibb, értelmezése egyszerűbbnek tűnik de a gépi kódú programozástechnika hamar bebizonyította, hogy a hexadecimális számok bevezetése nem szükséges rossz, gyakran nagy idő és energiamegtakarítás. A hexadecimális számrendszer 16 számjegyet különböztet meg, a 9-es számjegy után az A, majd sorban az A,B,C betűi következnek, egészen F-ig, vagyis F megfelel a decimális 15-nek, FF pedig 255-nek.

Látható, hogy 1 byte esetén maximum két karakterrel fejezhetjük ki a számot, míg két byte esetén mindössze maximum néggyel (65535=FFFF). Ez a memória közvetlen feltöltésekor nagy időmegtakarítást eredményezhet (a legtöbb MONITOR segédprogram lehetővé teszi a memória direkt feltöltését hexadecimális értékekkel).
Könnyen beláthatjuk, hogy a 8 és 16 bites számok értelmezése is sokkal egyszerűbb. A hexadecimális számjegyeket 4 bitenként külön csoportba sorolhatjuk, vagyis egy byte alsó négy bitje akkor lesz 1111, ha a hexadecimális érték F. Természetesen a felső négy bittel ugyanez a helyzet, így akár egy 16 bites cím képzése sem jelent problémát, pl.

```
|1 0 1 0|1 1 0 0| - |1 0 0 0|0 0 1 1|
    A       C           8       3
```

vagyis hexadecimálisan: AC83.

Decimálisan tudjuk, hogy az alsó byte értéke 131, a felső byte-é pedig 172. A 16 bites cím értékét így decimálisan a következőképpen számíthatjuk:

(172 * 256) + 131 = 44163

Ha tisztán hexadecimálisan dolgozunk, akkor megspóroljuk az ilyen jellegű számítást, mert a cím közvetlenül megjelenik előttünk.

Hogy fejthetjük vissza a hexadecimális számainkat decimális formába? A decimális számok képzéséhez hasonlóan a számítás helyiértékenként történik, jobbról-balra értelmezve, vagyis az adott helyiértéken álló számjegyet meg kell szorozni a számrendszernek azon hatványával, ahányadik helyiértéken az adott szám áll. Nézzünk erre egy konkrét példát.

3210 decimálisan a következőképpen bontható le:

( 3 * 10 * 10 * 10 ) + ( 2 * 10 * 10 ) + ( 1 * 10 ) + ( 0 * 1 ) = 3210

3210 hexadecimálisan a következőképpen bontható le:

( 3 * 16 * 16 * 16 ) + ( 2 * 16 * 16 ) + ( 1 * 16 ) + ( 0 * 1 ) =
( 3 * 4096 ) + ( 2 * 256 ) + ( 1 * 16 ) + ( 0 * 1 ) = 12816 (decimális)

(folytatjuk)

# EZ LESZ ITT AZ OLVASÓ OLDALA

*** Amennyiben igénybe veszi ***

Itt szeretnénk közrebocsájtani olvasói leveleket, észrevételeket, javaslatokat, itt hirdethet a kedves Olvasó, cserepartner kereshető stb. Várjuk klubok jelentkezését is, közölnénk címüket, a klubfoglalkozások rendjét, idejét, mert sokan nem ismerik az ilyen irányú lehetőségeket.
Egyenlőre - amíg nem kapunk visszajelzéseket - ismert angol software forgalmazók címeit közöljük. Tudomásunk szerint az illető software házak levél útján friss, naprakész információval tudnak szolgálni az általuk forgalmazott termékekről, ezen túlmenően színes reklámanyagokat és posztereket is küldenek az érdeklődőnek.

OCEAN & IMAGINE
6 Central Street
Manchester
M2 5NS
ENGLAND

THE EDGE
31 Maiden Lade
Covent Garden
London WC2E 8LH
ENGLAND

FIREBIRD
Wellington House
St Martin's Lane
London WC2
ENGLAND

VORTEX SOFTWARE
280 Brooklands Road
Brooklands
Manchester M23 9HD
ENGLAND

DIGITAL INTEGRATION
Watchmoor Trade C.
Watchmoor Road
Camberley
Surrey GU15 3AJ
ENGLAND

THOR
Erskine Industrial
Estate
Liverpool L6 1AP
ENGLAND

GREMLIN GRAPHICS
Alpha House
10 Conver Street
Sheffield S1 4FS
ENGLAND

MELBOURNE HOUSE
Castle Yard House
Castle Yard
Richmond TW10 6TF
ENGLAND

ACTIVISION
15 Harley House
Marylebone Road
London NW1
ENGLAND

US GOLD
Unit 10
The parkway
Industrial Centre
Heneage Street
Birmingham B7 4LY
ENGLAND

MASTERTRONIC
Park Lorne
111 Park Road
London NW8 7SL
ENGLAND

ELITE SYSTEMS
Anchor House
Anchor Road
Aldridge Walsall
West Midlands
WS9 8PW ENGLAND

A "Spectrum-Világ" szerkesztősége szeretné messzemenően az igények szerint összeállítani a kiadvány következő számait, ezért várjuk az Olvasó hozzászólásait, észrevételeit.

A TOP 10 lista egyenlőre szubjektív (a mi véleményünket tükrözi) de szeretnénk, hogy ha ezután a levélben beérkezett javaslatok alapján állna össze a sikerlista.

**Kéziratot nem őrzünk meg és nem küldünk vissza!**

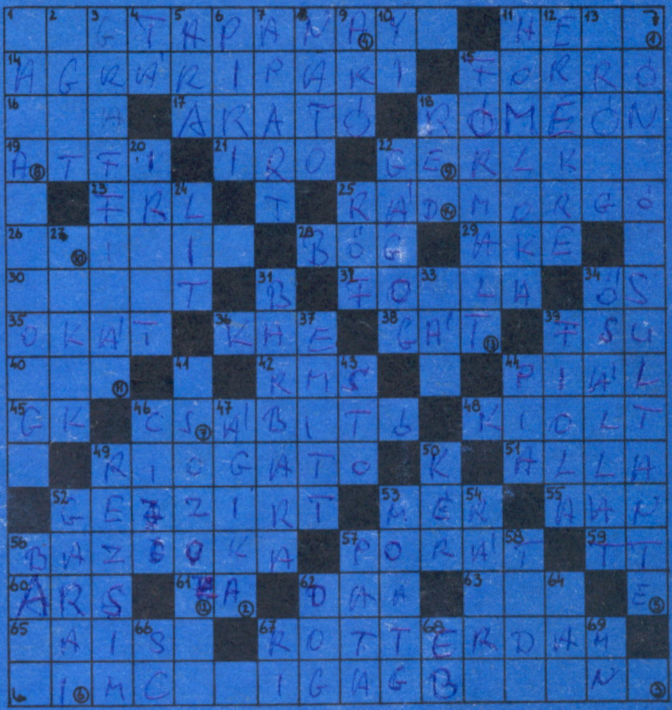
PÁLYÁZATI SZELVÉNY
OOOOO
OOOO
OOO
O

# Pályázati rejtvény

Ha megfejti az itt látható keresztrejtvényt, azoknak a mezőknek a tartalmából, melyeknek jobb alsó sarkában egy kis szám lett elhelyezve, egy három szóból álló szöveg olvasható össze, egy ismert angol hardware cég neve. Kérjük, hogy a megfejtést az alul látható - kivágható - pályázati szelvénnyel együtt, a megadott címre, zárt borítékban küldje el nekünk.
A beküldési határidő:
1987. október 10.

A helyes megfejtést beküldők között egy darab walkman-t sorsolunk ki (programjaink hallgatásához).

A nyereményt postázzuk.



Vízszintes: 1. Az ANDROID TWO c. program szerzője 11. 1986.év egyik legsikeresebb software-háza 14. A termelés túlnyomórészt mezőgazdasági jellegű 15. Nagyon meleg 16. A 6502 processzor egyik utasítása 17. Betakarító 18. Júlia párján 19. Magyar autójavító cég betűi keverve 21. Szerszáma a toll 21. Német község, magyar kiejtéssel 23. Francia, román és luxemburgi autók jelzése 25. Irántad nemtetszését kifejező 26. Érzelem 28. A szarvasbika teszi 29. Kicsinyítő képző, névelővel 30. Viselkedésével társait is erőszakra hangolja 32. A Spectrum legsérülékenyebb része 34. Előd 35. Előidézőjét 36. Fordított Egyesült Arab Köztársaság 38. Itt állnak a legények 39. Francia és szovjet autók jelzése 40. Lemezkezelő operációs rendszer 42. Némán rémes! 44. Iszik 45. Gépkocsi 46. Vonzó 48. Életet megszüntet 49. Ijesztgető 51. Aluminium és tükörképe 52. Melegvízforrást! 53. Súlyáról meggyőződik 55. AAN 56. ...Bill (ismert Spectrum-játék) 57. Apró szemű anyagát 59. Játékvasút típusa 60. ...-poetica 61. Tetőanyag fele 62. Z-80 gépi kódú utasítás 63. Balta angolul 65. Ezt írja ki a kártya-program, ha emeljük a tétet 67. Holland kikötőváros

Függőleges: 1. Itt található Taunton angol városban a Durell software cég 2. Fejetlenül nem engedte el! 3. Bevonalkázás 4. Az egyik morze jel 5. Menyasszony 6. Piroska 7. Valamin kívül angolul 8. Észak Atlanti Szövetség 9. Román gépjármű-típus 10. York's Industry 11. Fejének része 12. A vér szállításához erre is szükség van 13. Hibás angolul 15. Alakit 18. ... Fox (az ULTRAVOX énekese) 20. Irént 22. Liba teszi 24. 3/5 űrmérték 25. Régi hosszmérték 27. Fekete termést hozó, gubós növények 31. Női név 33. Elnyelheti az embert 34. Őslény 37. Nem amott 39. Kis üveg, elsősorban gyógyszer tárolására 41. Kazah színész (elöl vezetéknevével) 43. Mi? oroszul 44. Sör, bor, pálinka 46. Becézett József betűi keverve 47. Becézett női név 49. Államrend 50. Az ember a boltban teszi 52. Ismert piac Budapesten 53. Magyarország és arcus tangens rövidítése 54. Időmérő eszköz ellenértéke fordítva 56. Ismert sport-játék szimuláció 57. Lóláb 58. Yamaha szintetizátor-típusok betűjelei 62. Kutya 64. Dilemma magánhangzói keverve 66. Sport Club 67. Sír 68. Európa Bajnokság 69. A mangán vegyjele

PÁLYÁZATI SZELVÉNY

39,- Ft